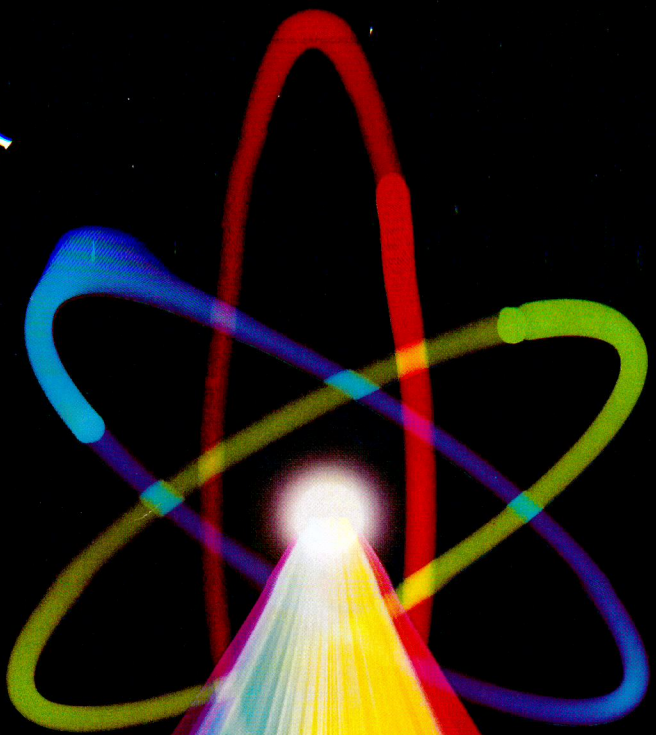
# RSTS PROFESSIONAL

**Volume 5, Number 1**

**February 1983**
$10^{00}$/issue, $35^{00}$/year


## INSIDE:

NOW AVAILABLE FOR VAX



# USER-11: POWERFULLY PRODUCTIVE.

People productivity. It's more important than ever. And a good database system can mean *real* productivity.

USER-11 is a high-performance database system.

It is a fact: Software designed with USER-11 is built more quickly, operates more reliably, and performs better than other software techniques.

USER-11 is unique. It's easy to install. Easy to learn. And easy to apply. Adaptive tools and a standard approach ensure that maintenance is easier than ever.

A key to USER-11's success is its powerful, dictionary-based modules. Software developers simply describe and assemble these modules to create custom business packages—at an

Naturally USER-11 is supported with excellent documentation and a variety of training options for beginner to expert. Our commitment is to your complete satisfaction.

Whether you are a software provider or a software user, we guarantee you will be delighted.

Ask us about USER-11 and our family of business software products, or better yet, ask a *productive* USER!

NCCS USER 11

**North County Computer Services, Inc.**
2235 Meyers Ave.,
Escondido, California 92025
(714) 745-6006, Telex: 182773

*USER-11 is currently available for DEC computers

# *What's the most efficient way to distribute financial models from DEC* to desks?*

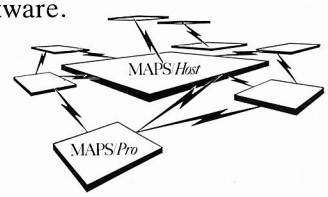
# MAPS/*Host*™ & MAPS/*Pro*™

* DEC, VAX *and* PDP *are registered trademarks of Digital Equipment Corporation.*
MAPS, MAPS/*Host and* MAPS/*Pro are registered trademarks of Ross Systems Incorporated.*

MAPS™ financial modeling software lets you offer your decision makers a single, comprehensive solution to their complex financial processing needs.

MAPS/Pro software is designed to run on DEC's powerful new Professional 350 desktop computer—to give you the full benefit of the 350's P/OS multi-tasking operating system, 5MB Winchester disk, bit-mapped graphics, special function keys and application menus. Fully-compatible MAPS/Host software runs on PDP-11 and VAX computers.

Working independently at desktop 350s, MAPS lets you and your users take advantage of state-of-the-art microprocessing hardware and software.




With 350s linked together, MAPS' common planning language and centrally-administered data base allow users to exchange data and financial models directly. Plus users can upload applications to a PDP-11 or VAX when additional computing power is needed—making *truly* distributed financial planning possible.

Designed for total financial decision support, MAPS features unlimited logic, professional report formats, sophisticated consolidation capabilities and advanced data calculations. Plus instant access to a common data base and model library that maximizes user efficiency. With MAPS there's no limit to the size or complexity of the system you can develop. Yet MAPS' online Help, Business English, and visual data editing make MAPS easy to use. And, fully-documented, it's easy to learn. What's more, Ross "Hot-line" support is as close as the phone if a problem does occur.

For more information on MAPS software, just return our coupon. Or call Ross Systems toll free at (800) 547-1000 (in California, call (415) 856-1100).

*OK, ROSS*

*R2*

*... I'd like to know more about how MAPS can offer me true distributed financial planning.*

*Please send more information on* ☐ *MAPS/Host* ☐ *MAPS/Pro*

☐ *I/We have access to a VAX or PDP-11*

☐ *Have a Ross representative call on me*

*Name* _____

*Title* _____

*Company* _____

*Address* _____

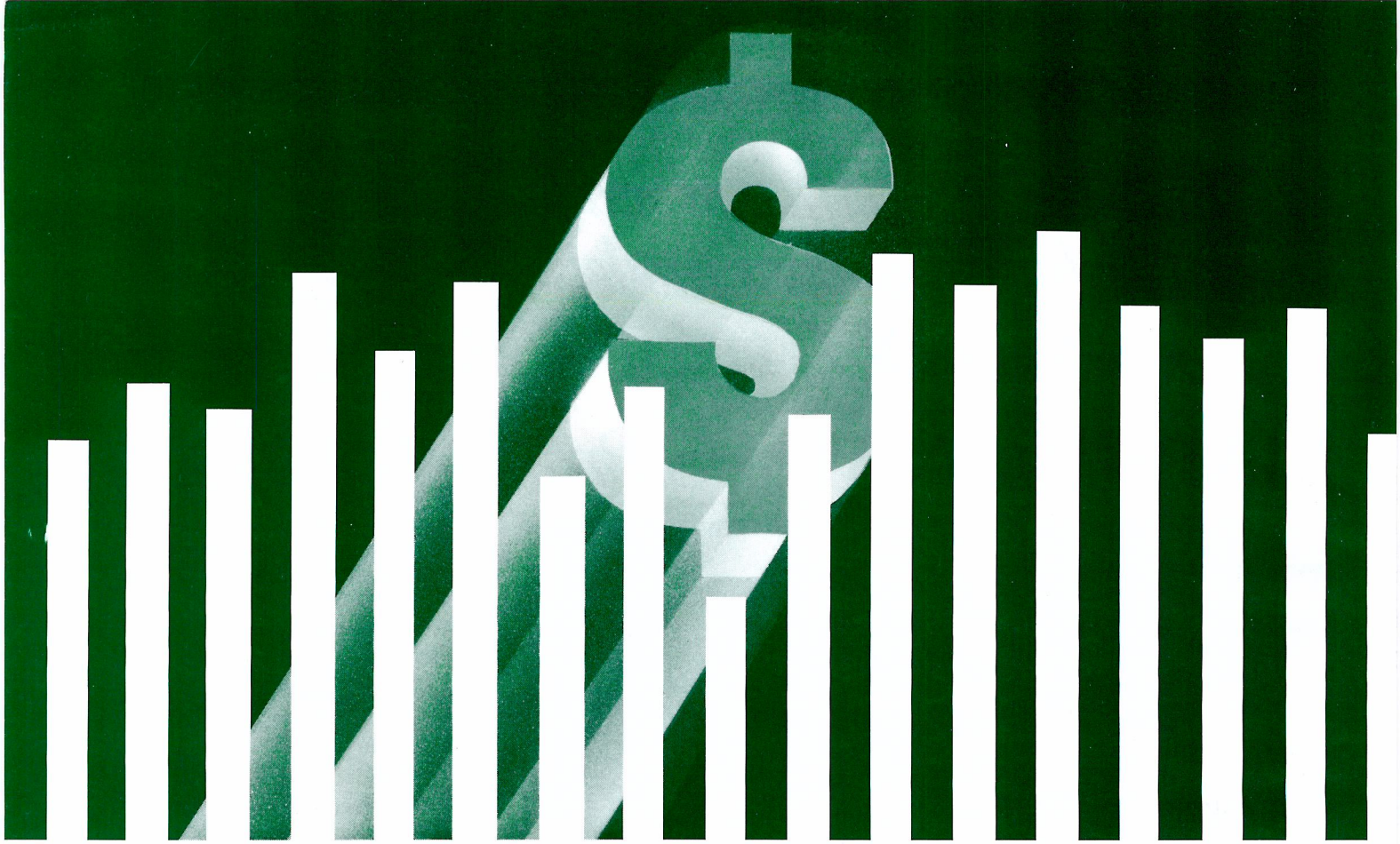
*City* _____ *State* _____ *Zip* _____

*Phone* (_____) _____

CIRCLE 1 ON READER CARD

ROSS SYSTEMS

*1860 Embarcadero Road*
*Palo Alto, CA 94303*
*Regional offices in San Francisco, New York, Dallas, Los Angeles.*



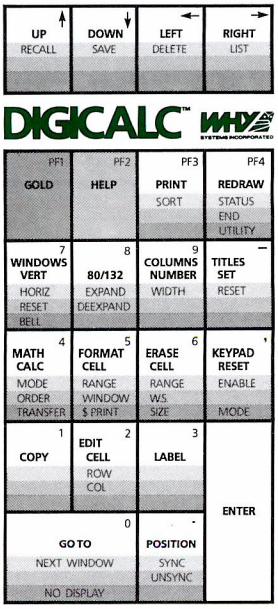
# DIGICALC,™ Powerful Financial Forecasting

With incredible range and depth, DIGICALC can provide the executive, accountant or professional with a worksheet capable of multi-year forecasts, budgeting and consolidations.

DIGICALC is designed for use on DEC* systems exclusively. The significance of its design is the incomparable degree of "help" built into it. DIGICALC has been called "the most user friendly program available for DEC computers."

DIGICALC runs on VMS*, RSTS/E*. More information is available in brochure form, but to really *feel* the power of DIGICALC call today for a free dial-up demonstration.

- FINANCIAL MODELING.
- ON-LINE HELP AND SELF TEACHING MODE.
- TEN KEY NUMERIC DATA ENTRY.
- EXTERNAL FILE INTERFACE.
- "BOARDROOM QUALITY" REPORTS.
- EXTENSIVE MATH FUNCTIONS ALGEBRAIC FUNCTIONAL LOGICAL SCIENTIFIC USER DEFINED FUNCTIONS.
- SAVES AND RECALLS WORKSHEETS.
- REGRESSION FOR FOUR PERIOD WITH BI-VARIANCE AND NINE PERIOD WITH FIXED VARIANCE.
- DYNAMIC NUMERIC AND ALPHABETIC SORTING OF ENTRIES BY ROW FOR AN ENTIRE WORKSHEET.
- CONSOLIDATION OF ANY NUMBER OF WORKSHEETS INTO A SINGLE SUMMARIZED OUTPUT WORKSHEET.





SYSTEMS INCORPORATED

16902 Redmond Way
Redmond, WA 98052 U.S.A.
(206) 881-2331


*DEC, VMS, RSTS/E are registered trademarks of Digital Equipment Corporation.



CIRCLE 72 ON READER CARD

# Contents



## Coming . . .

- DIBOL and MACRO
- QUEDEV.B2S
- RSTS/E Background
- Compilation Aid for Basic Plus Two
- CALC.BAS
- Resident Library Tips for Dibol Users
- Sort Benchmarks on DEC 11/34 with RSTS/E Report
- LSTLOG.MAC
- RSTS Resource Accounting
- MEMO.BAS
- The Rainbow 100: A First Look
- Fiber Optic Communications
- More . . .



The RSTS Professional Magazine, February 1, 1983, Vol. 5, No. 1. ISSN 0745-2888 is published bi-monthly by M Systems, Inc., 161 E. Hunting Park Avenue, Philadelphia, PA 19124. Subscriptions: single copy price $10^{00}$, $35^{00}$ per year; $50 Canada and 1st class. All other countries, air mail, $60 US. Second Class postage paid at Philadelphia, PA. POSTMASTER: Send all correspondence and address changes to: RSTS Professional, P.O. Box 361, Ft. Washington, PA 19034-0361, telephone (215)542-7008.

# From the editors. . .

### Whither RSTS

Dave Mallery

I hear that DEC plans to build about 2000 personal computers a week during the first year of production. It occurs to me that in a few years, the MAJORITY of DEC users will be personal computer owners. I know that I will certainly be one of them. How awful to become a DEC user and find yourself trapped in the CPM wasteland — perhaps never even find out about the operating systems that made DEC what it is . . .

I saw a '350' a few weeks ago. It was underwhelmingly slow. It seems that the Winchester drive is the villian. Apparantly, its seek time is in the same range as the floppy.

Much of the software currently being sold for micros seems to work its way around the 'speed' of the floppies by limiting the applications to the amount of data that can be held in memory. In many cases, the AMOUNT of data that can be handled is the severe limiting factor for the packages.

A TKB that runs at floppy speed is not really possible. In fact, most of our RSTS software relies on reasonably fast disc access. That leaves the '350' out. Not the '350' but its disc. There are lots of small fast Winchesters around. Lets hope one makes the grade soon.

I guess I am looking for a Personal RSTS machine with a '350' price. I don't want a run-only OEM box, but one that will let me write real code in real languages (B+2, v2?), compile and taskbuild in a time frame that makes the machine usable by a professional programmer.

Today's Wall Street Journal has an article on the new Apple Lisa system (16 bit, $9995). The wave of the present seems to offer lots of free software (a la Osborne), including word processing and spread sheets, preferably neatly integrated. Maybe we will be seeing some drastic reductions in DecWord and Dec-Calc soon.

I read that IBM, after they announce their new low-end PC, will have a multi-user engine in the works. For the life of me, I can't imagine that the machine I want has to be any more expensive than the '350'. The future of personal computing will be written by IBM and DEC. The market does NOT have to be driven by IBM.

I am betting that in a few years, the RSTS Pro will have an audience we never dreamed about back in Vol. 1, #1.

### Why Did It Die?

Carl B. Marbach

The little 11/40 was dead. No lights, no fans, no noise. The three RP02's were all in a line. Quiet. No familiar click,click,click from the disks as the heads were positioned by that clicking mechanism. There were terminals; 4 VT52's and one VT100. Terminals can't communicate with an unplugged computer. The room was dark but neatly piled in one corner was the residue of an active computer site: diagnostic tapes, hardware manuals, a RSTS V7.0 distribution kit, software documentation, backup listings and disk packs. I was quiet, reverent as if I were visiting a bereaved family. The 11/40 had died.

Why did it die? Don't 11/40's usually grow up and become 11/44's or 11/70's? O.K., some of them metamorphose into VAX/VMS systems. This one died and was REPLACED by another brand. Cables hung loosely from the ceiling and it felt like this room belonged more in an adventure game or in the dungeon . . ." you're in the computer room, it is dark and the computer is turned off . . ."

Was the 11/40 a bad machine? Did DECservice not respond when needed? Did the CPU get overburdened and run too slowly? Did the DEC salesman forget this account? I would guess that none of these caused the demise of this system. It probably died of software rigor softus a malady demonstrated to me one day when a major DEC OEM showed me his IBM system. "I shopped for software," he told me, "and this is what it runs on." Can you picture this DEC shop with an IBM system doing its computing. Ruined my day. I even like the OEM, they are down to earth good people with lots of good ideas. I respect their judgement and it worries me. Is the state of our software so bad, that this 11/40 dies and an OEM can't even use DEC equipment?

When I was in school they used to tell us that some day software would cost more than the hardware it ran on. Hard to believe then, hard now to imagine that it was ever not the case. Hardware continues to offer more bang for the buck while programmers get more bucks and seem to have less and less bang. What this all means is that software is gaining in importance every day, and it is already the most important part of any system. When a system dies, replaced by another, it is usually because of better software on the new machine. The 11/40 is a good machine, and when it runs out of power there are lots of good DEC replacements like the 11/44 or the 11/70 and even the VAX.

The beginning of the year is a good time to examine ourselves personally and professionally. How is your software? Is your system paying its way and providing value to your company. Can you make it do better? If we all work just a little harder to make sure our software systems are all they should be we could insure our systems continued success and valuable contributions to the companies we work for.

Remember the users too, be sympathetic — they have problems also. Keep those lights (if you have them) blinking, the disks clicking, the printers printing, the plotters plotting, the tapes spinning and the terminals typing.

I don't like quiet computer rooms. ♥

RSTS PROFESSIONAL

Publishers
R.D. Mallery
Carl B. Marbach

Managing Editor
James L. Trichon

Director of Advertising
Helen B. Marbach

Business Manager
Peg Leiby

Assistant to the Editor
Hope Makransky

Editorial Assistant
Linda DiBiasio

Subscription Fulfillment
Kathi B. Campione
Claire Hollister

Steven Barsh
Margie F. Pitrone

United Kingdom Representative
Pauline Noakes
RTZ Computer Services Ltd.
P.O. Box 19, 1 Redcliff Street
Bristol, BS99-7JS
Phone: Bristol 24181

Mid-Atlantic Representative
Ed Shaud
P.O. Box 187
Radnor, PA 19087
215/688-7233

N.Y. & New England Representative
Jack Garland
P.O. Box 314 S.H.S.
Duxbury, MA 02332
617/934-6464

Contributors
Rob Frazer
Paul A. Gilberti, Jr.
Philip Hunt
Michael H. Koplitz
Alton O. Moore, III
Niall McPhillips
David Spencer

Lawrence P. Gallagher
Michael C. Greenspon
Greg Justice
Michael Mayfield
Charles W. Mustain
Allen Rueter
Allan Woloshin

Cartoons
Douglas Benoit
Emily T. Dahlstrom

Cover Photo
David Sheppard

Design & Production
Grossman Graphics

Printing & Binding
Schneider Litho Co., Inc.



Editorial Information: We will consider for publication all submitted manuscripts and photographs, and welcome your articles, photographs and suggestions. All material will be treated with care, although we cannot be responsible for loss or damage. (Any payment for use of material will be made only upon publication.)

# BEFORE you add memory (or anything else) to increase system performance


# You should add DOPTER!

DOPTER is an easy to use RSTS/E disk copying program which

**INCREASES SYSTEM PERFORMANCE UP TO 50%.**

DOPTER performs all of the standard functions necessary to structure a RSTS/E disk volume and automatically does the following:

- Places all files and free space in their optimum positions.
- Produces better optimized MFD/UFD's than REORDR.
- Deletes unused file attributes from source, task, and object library files saving UFD and cache accesses.
- Places and pre-extends the MFD.
- Places the most used files at the front of the UFD's.
- Places the UFD's with the most activity toward the front of the MFD.

## For More Information

If you would like more information on how you can increase the performance of your RSTS/E system with DOPTER and a free copy of "RSTS/E DISK OPTIMIZATION IN A MULTI-USER ENVIRONMENT", phone or write SPH today.

RSTS/E is a registered trademark of Digital Equipment Corporation.

SPH

System Performance House, Inc.

5522 Loch More Court · Dublin, Ohio 43017 · 614-265-7788

CIRCLE 108 ON READER CARD

# LETTERS to the RSTS Pro...

Send letters to: Letters to the RSTS Pro, P.O. Box 361, Ft. Washington, PA 19034-0361.

To start off, I would like to thank you for a really terrific magazine. It has helped me with my system, and it is extremely interesting! Whenever I get my RSTS PRO, it is always the high-point of my week.

Next, I think that by far the best thing yet to come from M SYSTEMS is the RSTS/E MONITOR INTERNALS manual. I say this before I have even seen it. Enclosed, please find my order for the RSTS/E Internals Manual. I have a question — the second paragraph tells of "Future Updates." Can you give us an idea (just a guess, if you don't yet want to make a commitment) when (or how often) these updates will be available, and how much they will cost?

Also, I read the letter from Mike Mayfield, explaining why RSTS/E jobs can only use 31K. One of the more exciting things I have read so far in your magazine, is that he might offer a patch to allow 32K in Version 7.1! I hope he can show us how to do this in V7.2 also! Thank you, Mike!

Next, I would like to point out some errors I spotted in the August issue of RSTS PRO.

In DEAR RSTS MAN, a letter from SPIDL details the woes of stopping a line printer spooler at night, and trying to allow non-privileged users to start it up again in the morning. RSTS MAN suggested using QUE-11 V2.2. May I suggest two less expensive methods?

First, to do exactly what SPIDL asked for, you could make a modified version of the program SPOOL (call it SPOOLY). You would modify the code to not do any input from the terminal, but to act as if a proper command had been entered to become LP0: (or whatever). You also might include code to log out, then log in to [1,2] *AFTER* it had detached. Then you could compile SPOOLY with a protection code of 232. To start up the spooler in the morning, the non-privileged operator would merely RUN $SPOOLY. SPOOLY would declare itself a receiver, as SPOOL does, detach, as SPOOL does, and chain to SPLIDL, as spool does. I leave it to the reader to discover a method for shutting down the spooler at night.

An even better method, though, is not to shut down the spooler at all! Just pick an account to become an "operator account," and tell OPSER about it: PL/OPER KB0:[p,pn]. Then, in order to shut down the spooler at night, type (assuming this is LP0:) PL/INT LP0SPL:FORM OFF which tells the spooler to handle only jobs queued to form OFF. If you don't queue anything to FORM OFF, nothing will be printed! In the morning, type PL/INT LP0SPL:FORM NORMAL which tells the spooler to handle only jobs queued to form NORMAL. This is the form queued to if no /FORM switch is used on the QUE command. If using this method, I would suggest using patches 14.4.1F, 14.6.3F (which prevent the operator from QUEING to someone else's account), and patch 14.6.1F (which eliminates the "JOB WITH DIFFERENT FORM NAME WAITING" message; if this patch is not installed, then anytime something wakes up QUEMAN during the night that message will be printed). In another letter to DEAR RSTS MAN, Jim Carrigan asks how to create a Tempfile (.TMP file) which will automatically be deleted by LOGOUT. It is really sad that DEC technical support could not help him. This is really simple. It does not involve any special bits in the UFD, and no RSTS MAGIC is needed. (There is a function of the CALFIP directive which does some of this for you, but it cannot be used from BASIC or BP2 and besides, you don't need it.) Simply name the file with any four letters, followed by your job number, with the extension (excuse me, filetype) .TMP. For instance, a program called FROG using a tempfile should call it FROG02.TMP, if job 2 is running. It must be a 2-digit job number, even if it is job 9 or less. If you never CLOSE the file for the duration of the program, and you do not need it again on subsequent runs, you can KILL the file immediately after opening it. Then, as soon as the program ends or the file is closed, the system purges it. (You can use a KILLed file as if it were not killed. The system remembers that it has been "Marked for Deletion," and the file is killed as soon as it is closed.) If for some reason you cannot do this, LOGOUT will kill the file for you when the job logs out (note: the file is not deleted if the job is KILLED). If I am job 2, LOGOUT does the equivalent of PIP ????02.TMP/DE as it is run. (To check this, go into and out of TECO, do a directory, then type BYE/N and do another directory.)

I don't mean to pick on RSTS MAN, but another letter, from Jerry Forshee, deserves a note about upgrading PDP 11/34 systems to PDP 11/44 systems USING THE SAME COMPUTER SYSTEM. This is an option that most DEC salespeople do not seem to know about! Simply remove the 2 PDP 11/34 cards, the FP-11 (if any), the DL11, and the memory. Install in its place the single PDP 11/44 card and some PDP 11/44 memory. That's it—you don't even need a new SYSGEN! The entire procedure, including running diagnostics, takes about 3 hours, and there is almost nothing that CAN go wrong. Another possibility, for those that cannot afford the PDP 11/44 CPU, is ENABLE/34 (by Able Computers). See my article *["Able Computer Technology" p. 28, this issue]* for more details about this.



There is something funny about OSCAR.RTS (from MACRO MAN, August, 1982). Every time I run the MAKSIL program, I get the error ?Partition or stack parameter incorrect for task ?Task image OSCAR.TSK cannot be converted to run-time system OSCAR. I tried this using both the "EDIT MODE" of MAKSIL and using OSCAR.TEC. I also tried changing the ".PSECT" commands in OSCAR to ".PSECT .99998" and ".PSECT .99999" and changing the EXTSCT command with the task-builder to EXTSCT$.99998:0 (or EXTSCT$.99998:17516). Did I do something wrong?

Next, I would like to say that I agree with Steven Edwards [Letters, August, 1982]. It is one thing to see an article discussing things that should be available on RSTS/E, and using an existing software product as an example. But it is quite another thing to see a user's manual, including an installation guide (!), appearing in the guise of a RSTS PRO article! You two run a terrific magazine. It is a pity that such articles, with such a limited interest (not all RSTS users can benefit from them, only those who buy the products), should be included with the rest.

Unfortunately, there is also the grey area. Articles like TYPE (August, 1982) are of some interest to me, because now I have an idea for a program I can write myself for my own system. Only later did I notice that the TYPE program is listed for sale in the classified ads. I found the TYPE article interesting. I might not have if it had been 10 pages instead of 1.

So I have not written. After all, where does one draw the line? I DO have a big enough mouth to tell you how to run your magazine, if I think you are doing it wrong. Unfortunately, I cannot even think of a reasonable suggestion to make, when it comes to drawing this line. I guess that's why you guys print the magazine, and I only buy it. You get the headaches. I CAN say that I hope you never do what DECUS is doing — banning all talk about anything non-DEC. If you had this policy, I wouldn't know what ROSS/V is, or a D-MAX, or a SUPER-MAX, or ENABLE/34, or . . . etc. It has finally dawned on me what you both have been saying all along that WE, the READERS, are the SAME PEOPLE as the contributors. You would not publish an article on how to run QUE-11 if you had some other material which you judged to be better. No longer will I say, "Gee, I hope he gets some more people to contribute," until I have contributed myself. In that vein, find enclosed a copy of an article about the pros and cons of ENABLE/34. [I don't know if you remember, Dave, but I promised this to you at L.A. DECUS 1981! I was the guy wearing the badge with the homemade banner that said, "NOBODY."]

I am not Guru-ish; I might have my moments, but they are the exceptions. But not all your readers are 10-year RSTS hackers! Tell me if you would like an article for beginners at system management. I might be able to tell some of the simpler (DEC-supported!) things that can be done to make a system flow smoothly, quickly, and securely. Would you be interested?

And, in case my letter gets published, let me say to all of you in reader-land: THINK! What have you done lately on RSTS? If it was easy and it worked well, write in to tell us all to do it, too! If it was hard but paid off, write in to tell us why it was hard, and how to make it easier! If it

. . . continued on page 31

Intersil Systems

# MEETS THE DEC® DH-11 CHALLENGE

The challenge; increased communications capacity, increased reliability, and improved technology – at decreased cost.

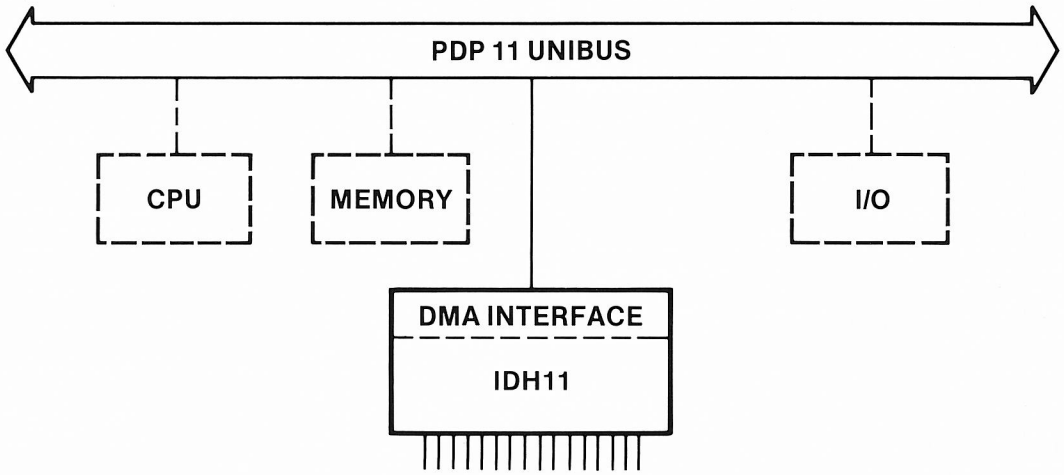
## That's not a challenge for Intersil Systems.

- **Increased Capacity –** One Intersil DH-11 replaces nine DEC card slots
- **Increased Reliability –** Only one chance for a failure instead of nine.
- **Improved Technology –** The latest microprocessor based technology is employed.
- **Decreased Cost –** 66% less! One Intersil DH-11 costs 66% less than one DEC DH-11.

**In addition the Intersil DH-11 offers these features:**

- 16 Asynchronous local or remote channels on the UNIBUS® .
- DMA output to free the CPU from Interrupt handling.
- On-board diagnostics.
- Complete software compatibility.

## IDH11




**16 ASYNCHRONOUS CHANNELS TO LOCAL OR REMOTE TERMINALS**

Get all the challenging facts. Call (408) 743-4300, TWX: 910-339-9369, or write Intersil Systems, Inc., 1275 Hammerwood Avenue, Sunnyvale, CA 94086

® Trademarks of Digital Equipment Corporation

CIRCLE 171 ON READER CARD

# MAKING RSTS REALTIME

By Michael Mayfield,

Northwest Digital Software, Box 2-743, Newport, WA 99156

RSX is a realtime system and RSTS is a timesharing system and never the two shall meet, right? Wrong. This article provides a patch to the RSTS monitor that will give you all the realtime response of RSX while still providing the timesharing we know and love on RSTS.

Although a true RSX affectionado probably wouldn't agree with me, RSTS and RSX really aren't that different any more. While RSX has become more "friendly," RSTS has been getting more technical tools. But, even with these increasing similarities, one big difference still separates them: RSX can provide realtime response and RSTS cannot.

By realtime response I mean that whenever a program wants to run that has a priority higher than the program currently running, the current program will be immediately suspended and the higher priority program will be run. The key word here is "immediately."

Response time in RSTS is tempered by the run burst associated with each job. Once a program starts running, it can continue to run until it has used up this run burst, even if a higher priority program becomes runnable. This means that, statistically speaking, average realtime response in RSTS is one half the average run burst, or 1/20th second. Not shabby, but definitely not realtime.

The patch described below provides RSTS with the same realtime response as RSX for programs of sufficient priority while retaining normal time sharing response for programs of lower priority.

When a program completes an I/O request it will immediately start running if it is within the realtime priority range and is of a higher priority than the program that is currently running. Average response time is less than one millisecond, even on an 11/23.

The only drawback to adding realtime processing to RSTS in this way is that normal timesharing can become skewed by frequent interruptions for realtime processing. When the currently running program is suspended so that the realtime task can execute, the current program will lose its runburst and any other runnable programs of the same priority will be scheduled before the current program runs again.

In the patch below, the realtime processing priority range is specified by the number at offset 40. Any program at or above the priority specified by this value will be treated as a realtime task. Any program of a lower priority will be treated as a normal timesharing task.

The realtime processing range is currently set to priority 64 and higher. This threshold can be changed by entering a different value at offset 40.

This patch uses patch space that is allocated for possible monitor patches. Future monitor patches may require the same patch addresses. If this occurs, the realtime patch will have to be removed or moved to a different location in patch space. The patch is position independent and can be installed in any other area in patch space that is not in use.

The comments following the semi-colons are for information only and can be ignored while entering the patch, although they will not cause any problems if entered. <LF> is used to signify a linefeed. ?????? is used to signify that any value is acceptable for this field.

As with all patches, be sure that the offset and old values are correct for each line before making any changes. If any of the old values are incorrect, abort the patch by typing ↑C.

```
RUN [1,2]ONLPAT
Command file name? RELTIM.LOG=
File to patch? <LF>
Module name? <LF>
Base address? PATCH+300
Offset address? 0
Base    Offset   Old      New?
??????  000000   000000   ? 105737  ;Is a job currently running
??????  000002   000000   ? JOB
??????  000004   000000   ? 1423    ;No- Always call scheduler immediately
??????  000006   000000   ? 16446   ;Point to JDB for job completing I/O
??????  000010   000000   ? JOBTBL
??????  000012   000000   ? 62716   ;Point to JDB+JDPRI (job's priority)
??????  000014   000000   ? 34
??????  000016   000000   ? 13746   ;Point to JDB for current job
??????  000020   000000   ? JOBDA
??????  000022   000000   ? 62716   ;Point to JDB+JDPRI (job's priority)
??????  000024   000000   ? 34
??????  000026   000000   ? 123676  ;Is current prio >= I/O job's prio
??????  000030   000000   ? 0
??????  000032   000000   ? 2007    ;Yes- Don't do realtime scheduling
??????  000034   000000   ? 127627  ;Is I/O job's prio >= realtime threshold
??????  000036   000000   ? 0
??????  000040   000000   ? 64.     ; (realtime priority threshold)
??????  000042   000000   ? 2403    ;No- don't do realtime scheduling
??????  000044   000000   ? 52737   ;Schedule the I/O job immediately
??????  000046   000000   ? 20000
??????  000050   000000   ? L3QUE2
??????  000052   000000   ? 5726    ;Junk pointer to JDB for I/O job
??????  000054   000000   ? 105737  ;Replace the patched instruction
??????  000056   000000   ? JOB
??????  000060   000000   ? 207     ;Return from patch
??????  000062   000000   ? ^Z
Offset address? ^Z
Base Address? IOFIN4
Offset address? 62
Base    Offset   Old      New?
??????  000062   105767   ? 4737    ;Enter patch
??????  000064   ??????   ? PATCH+300  ;NOTE: New Value must match patch base
??????  000066   001003   ? ^C      ;End of patch
```

# THE HOUSE of VAX

Under one roof, Hamilton provides *all* your VAX needs. Here's how:

**RENTAL** Latest VAX Systems for 6 to 24 month rentals, with full upgrade flexibility, purchase option and prompt delivery.

**TIMESHARING** Extensive VMS based library of DEC layered products, application packages and software tools.

**RSTS / RSX to VMS** Conversion Services featuring fully supported PDP11 operating systems for media and software conversion.

**SOFTWARE LIBRARY** Including Word Processing, Data Base Management, Business & Engineering Graphics, Accounting, System Management and Accounting, Container Optimization, COBOL Program Generator.

**HARDWARE CONFIGURING** All DEC systems and peripherals together with Hewlett Packard and Tektronix graphic devices, Dataproducts and Diablo printers, pretested and installed on your site with manufacturers warranty and maintenance available.



**HAMILTON**
**6 Pearl Court, Allendale, N.J. 07401**

Please rush me information on the following:
(Circle and/or fill out items below)

**RENTAL** 730/750/780 PDP11 _____
**TIMESHARING APPLICATION** _____
**CONVERSION** RSTS RSX RT11 CTS
**SOFTWARE** _____
**HARDWARE** _____

Name _____
Position _____
Company _____
Address _____
City _____ State _____ Zip _____
Telephone Number _____

**HAMILTON**

HGL Software Hamilton Rentals

6 Pearl Court, Allendale, N.J. 07401
**TOLL FREE 800-631-0298**
In New Jersey 201-327-1444

415 Horner Ave., Toronto, M8W4W3
416-251-1166
**TOLL FREE**
**Ont. & Que. 800-268-2106**
**All Other Prov. 800-268-0317**

NEW YORK • DALLAS • MONTREAL • CALGARY
LONDON • PARIS • DUSSELDORF

CIRCLE 10 ON READER CARD

# THE RSTS CRYSTAL BALL

## RSTS V7.2 Enhancements

By Michael C. Greenspon, Integral Information Systems, Los Angeles, CA



Welcome once again. In this issue, I have some interesting monitor enhancements, as well as a fix for the named directory code.

### ERRATUM

First, some corrections for last month. A small but serious error found its way into the TECO macro (PI.TEC) during publication. The sixth physical line of the macro, as printed, should end with the instructions -1%I, followed by a close iteration, ">". The printed version showed the ">" replaced by a paragraph symbol, which is a little difficult to key on an ASCII terminal.

Also, the load average code listed at the end of the article was a little munged. The code is supposed to start with the second .ENABL LC. Only the disclaimer got crunched — the code is all OK. Also, note that the monitor load average computation routine ends with the .END, and the block of code after it is a BASIC program intended to print out the current system load averages. These two were run together without any spacing, making it hard to tell where one ended and the other started.

### MONITOR ENHANCEMENTS

I assume by this time that most sites are running version 7.2. All monitor patches and enhancements I publish for V7.2 should work for V7.1 as well, unless noted.

Two patch locations in FSS may be used to enable a useful feature. When enabled, the patch allows a project,programmer specification in the form "[,n]" and "[]"; i.e., omitting the project number or the PPN completely. If the project number is omitted, it defaults to the project number of the account which the calling job is logged into. If the entire PPN is omitted; i.e., empty brackets, it will default to the account that the calling job is logged into. For example, a job logged into [100,100] executes a .FSS call on the string "[,5]". With the patch enabled, the call will return [100,5] as the PPN, instead of an error. The string "[]" would return [100,100]. Note that since .FSS does the translations, some programs which parse their own PPNs will not accept these specifications.

The patch (actually two separate patches) is listed below. If patched on-line into the installed SIL, it will not take effect until the system is rebooted.

```
File to patch?
Module name? EMT
Base address? ..PJDF
Offset address? 0
 Base    Offset  Old     New?
??????  000000  000405  ? BNE!(Q&377)   ; Make into a BNE to enable [,n]
??????  000002  016703  ? ^C            ; Up-arrow C to exit

File to patch?
Module name? EMT
Base address? ..PPDF
Offset address? 0
 Base    Offset  Old     New?
??????  000000  000410  ? BNE!(Q&377)   ; Make into a BNE to enable []
??????  000002  016703  ? ^C            ; Up-arrow C to exit
```

I have finally disassembled the named directory code, and, after staring at it for nearly an hour, discovered several bugs. Unfortunately, none of these bugs is the MFD-bonking type reported by several sites running named directories. However, I have been running a monitor with named directories all weekend, making heavy use of the FSS and UU.NME translations, and have not been able to bonk any of the packs. I even wrote a program to log in and out and read random blocks of the MFD on the system disk, hoping to find some weird FIP condition that would punt garbage in FIBUF back to the MFD. Alas, three days later, the MFD is still quite readable — and the named directory code appears to work just great!

I am rating the use of the named directory code as "safe." Our development machines are running it now, on the normal production packs, and I am reasonably confident that we won't have any problems. My patches fix some serious bugs, however these bugs would at worst cause some confusion or frustration on the part of the user — no random disk writes. Obviously, I cannot take responsibility for anyone munging a pack with the named directory code. If you do decide to run it, and you manage to bonk a pack with it, please, please contact me. I'd like to find some REPRODUCIBLE condition which causes the code to fail.

The RSTS named directory implementation is a "slap it on top of what we've got" job, but a rather neat and elegant one. It is based on a small hashed file called NAME.SYS, located in [0,1] on all disks with named directories. The file contains enough information to translate 12 character ASCII (4 words of RAD50) names to PPNs and back.

The first block of the file is used as a directory/scratch block, and stores the hash table, links to the PPN table and free blockette list, number and maximum number of entries in the file, etc. Another contiguous section of the file is used to store the PPN table, which contains one word for each

name entry. The final section of the file is used to store the name blockettes. The minimum size of the file is, therefore, two blocks. A file capable of storing 1024 names must have a size of (only!) 37. blocks (1 block of header, 4 of PPN table, and 32. of name blockettes).

The primary entry for an account contains the name returned by UU.NME for a lookup by PPN function. Alias entries are used to translate "aliases" for the account to a PPN. Each account on the disk may have only one primary entry, but any number of aliases.

Since NAME.SYS resides on each disk with named directories, the same account name may exist across disks. The name translation is disk specific, and directory names do not imply devices. Therefore, DR1:[FOO] and DR2:[FOO] are different accounts, as are DR1:[1,2] and DR2:[1,2]. Also, DR1:[FOO] and DR2:[FOO] do not have to reference the same PPN — DR1:[FOO] could be DR1:[1,2] while DR2:[FOO] is DR2:[100,100].

Names are translated to PPNs via a small hash table, which contains pointers to a linked list of name blockettes. The numeric result of hashing a name is used as a pointer into the hash table. The linked list is chased to find the matching name blockette, and the PPN (and a user definable word) are returned.

PPNs are translated to names by scanning through a table of PPNs. When a match is found, the address of the matching PPN is used as an offset into the name blockette portion of the file to retrieve the correct name blockette. Zero words in the PPN table indicate entries which aren't in use.

Free name blockettes are kept in a linked list, the head of which is pointed to by a word in the first block of the file.

Users may translate directory names to PPNs via the .FSS call (file name string scan). The UU.NME call is used to go the other way; i.e., PPNs to names, as well as to add and delete entries from NAME.SYS.

I have located four bugs in the named directory code. The first two are really the same bug, but in two different places. As a result of this bug, translations from PPNs to names fail under certain conditions. The third bug is a nasty fencepost error, which causes a few (random) words of monitor memory after FIBUF to be RAD50-unpacked and returned as a name. The fourth bug is an omission — the lookup code never validates the PPN, and the add/delete code doesn't check for wildcard PPNs.

The first bug works like this: The code to locate the passed PPN in the PPN table starts out at the top of the table, and with a count of the number of entries in the file. It steps through the PPN table, trying to find a match for the given PPN. Each time the PPN doesn't match the one in the table, the count is decremented. When the count becomes zero, the code assumes that all the entries in the table have been scanned, and returns a NOSUCH error.

This routine contains a serious logic error — it forgets to check for unused entries in the PPN table. If entries are added and deleted, eventually there will be holes in the table. The zero PPNs won't match the passed PPN, but the count will be decremented anyway. This means that some possibly matching entries will not be searched.

The trivial fix for this is to add a check for the entry being zero. This causes complications, however, since alias entries occupy a name blockette, but have zero entries in the PPN table. A lookup call on a file containing only alias entries would read past the end of the PPN table, and possibly past the end of the file. Equate that to bad-news in your symbol table . . . Anyhow, my fix is to set the entries in the PPN table for aliases to -1, so that they can be distinguished from unused entries, yet won't be matched during a search by PPN.

The resulting word offset into the PPN table is used as an offset into the name blockette portion of the file. The fencepost error is in the conversion from a word offset to a block and byte offset. For some reason, the code is subtracting 512. (a block worth of bytes) from the byte offset, and counting how many times it can do this without carry. (Hey guys, ever hear of a DIV instruction? Yeah, it does the same thing as your little loop.) Anyhow, the branch after the compare is a BLOS. It should be a BLO. If this routine attempts to translate a name blockette index which is a multiple of 32., (512./16. bytes/blockette = 32. blockettes/block) the byte offset into FIBUF will be an incorrect 512., and whatever is after FIBUF in memory will be used as the name blockette. Whoops! The fix for this is quite simple — the BLOS becomes a BLO.

Finally, I've added a little code to verify that the PPN passed to UU.NME is valid. If not, the call will return a BADNAM error.

The patch for UU.NME is listed below. Unfortunately, there was no patch space available at the end of the NME module. I've used RSTS PATCH for the fix. This may conflict with existing or future DEC patches for V7.2. If anyone has any problems, contact me for instructions on how to move my patch elsewhere.

Also, please note that this is a patch to both OVR and a resident module. The patch to the resident module may be installed either on or off-line, but MUST be installed first. The patch to OVR should be installed off-line. One person installed my UU.TRM patch on-line, and then wondered why TTYSET crashed his system. If you install an OVR patch, it takes effect immediately (or when you snap OVR.SYS, if you have one). Since the patch to the resident module doesn't go into effect until the system is rebooted, the OVR patch will jump into null patch space and halt the processor. If you're not sure of what you are doing, install this patch OFF LINE!

Patch to RSTS module for UU.NME. May be installed on-line. MUST be installed before OVR patch.

```
File to patch? <LF>
Module name? RSTS
Base address? PATCH
Offset address? 234
 Base    Offset  Old     New?
??????  000234  000000  ? MOV!6402       ; PATCH1: Get the passed PPN
??????  000236  000000  ? FQPPN
??????  000240  000000  ? BEQ!16         ;  none, default it
??????  000242  000000  ? CMP!2702       ; Is it [0,1]?
??????  000244  000000  ? 1
??????  000246  000000  ? BEQ!11         ;  yes, go on
??????  000250  000000  ? SWAB!02        ; Swap proj. and prog. number
??????  000252  000000  ? BEQ!6          ; Project was 0 and not [0,1] -- er
??????  000254  000000  ? CMPB!2702      ; Is project 255.? (wildcard)
??????  000256  000000  ? 255.
??????  000260  000000  ? BEQ!3          ;  yep, error
??????  000262  000000  ? SWAB!02        ; Swap the bytes back
??????  000264  000000  ? INCB!02        ; Is prog. number 255.? (wildcard)
??????  000266  000000  ? BNE!1          ;  no, go on
??????  000270  000000  ? TRAP!BADNAM    ; Error with ?Illegal filename
??????  000272  000000  ? ADD!2705       ; Skip defaulting of PPN on return
??????  000274  000000  ? 6
??????  000276  000000  ? RTS!05         ; Get out
??????  000300  000000  ? INC!05         ; PATCH2: Bump index into PPN table
??????  000302  000000  ? TST!61         ; Was last entry free?
??????  000304  000000  ? -2.
??????  000306  000000  ? BEQ!2          ;  yes, so don't decrement count
??????  000310  000000  ? DEC!00         ; Decrement the count of names
??????  000312  000000  ? RTS!07         ;  and get out
??????  000314  000000  ? CLZ            ; Make sure we don't take the branc
??????  000316  000000  ? RTS!07         ; Get out
??????  000320  000000  ? ADC!03         ; PATCH3: 2nd half of double prec.
??????  000322  000000  ? MOV!61C0       ; Get number of entries in file
??????  000324  000000  ? 10
??????  000326  000000  ? RTS!07         ;  and get out
??????  000330  000000  ? CMP!6421       ; PATCH4: does this entry match PPN
??????  000332  000000  ? FQPPN
??????  000334  000000  ? BNE!1          ;  no, go on
??????  000336  000000  ? RTS!05         ; Get out
??????  000340  000000  ? TST!25         ; Skip branch
??????  000342  000000  ? TST!61         ; Was last entry a free entry?
??????  000344  000000  ? -2.
??????  000346  000000  ? BNE!2          ;  no, just get out
??????  000350  000000  ? ADD!2705       ; Skip the decrement and branch
??????  000352  000000  ? 4
??????  000354  000000  ? RTS!05         ; Get back, get back, da da da...
??????  000356  000000  ? TSTB!64        ; PATCH5: is this a primary add?
??????  000360  000000  ? FQPFLG-FQNAM1
??????  000362  000000  ? BMI!4          ;  yes, go on
??????  000364  000000  ? MOV!2764       ; Insure that PPN is updated as -1
??????  000366  000000  ? -1
??????  000370  000000  ? FQPPN-FQNAM1
??????  000372  000000  ? RTS!05         ; Return
??????  000374  000000  ? TST!25         ; Skip branch
??????  000376  000000  ? RTS!05         ;  and return
??????  000400  ??????  ? ^C            ; Up-arrow C to exit
```

Patch to OVR for UU.NME. Should be installed OFF LINE!

```
File to patch? <LF>
Module name? OVR
Base address? NME0
Offset address? 202
 Base    Offset  Old     New?
??????  000202  005764  ? JSR!537        ; Call patch space @ PATCH1
??????  000204  000006  ? PATCH+234
??????  000206  001002  ? ^Z
Offset address? 370
 Base    Offset  Old     New?
??????  000370  005205  ? JSR!737        ; Call patch space @ PATCH2
??????  000372  005300  ? PATCH+300
??????  000374  001406  ? ^Z
Offset address? 504
 Base    Offset  Old     New?
??????  000504  101404  ? BLO!(Q&377)    ; Change BLOS to BLO
??????  000506  162705  ? ^Z
Offset address? 676
 Base    Offset  Old     New?
??????  000676  026427  ? NOP            ; Blast old PPN checking code
??????  000700  000006  ? NOP
??????  000702  000001  ? NOP
??????  000704  001404  ? NOP
??????  000706  105764  ? NOP
??????  000710  000007  ? NOP
??????  000712  001001  ? NOP
??????  000714  104422  ? NOP
??????  000716  004537  ? ^Z
Offset address? ^Z
Base address? ^Z
Module name? ^Z
File to patch? ^Z                       ; It is important to Z back to this
File to patch? <LF>                     ; question!
Module name? OVR
Base address? NME1
Offset address? 154
 Base    Offset  Old     New?
??????  000154  005503  ? JSR!537        ; Call patch space @ PATCH3
??????  000156  016100  ? PATCH+320
??????  000160  000010  ? BEQ!17         ; No names, so no PPN table scan
??????  000162  004737  ? ^Z
Offset address? 172
 Base    Offset  Old     New?
??????  000172  026421  ? JSR!537        ; Call patch space @ PATCH4
??????  000174  000006  ? PATCH+330
??????  000176  001421  ? BR!(Q&377.     ; File exists, error
??????  000200  005300  ? ^Z
Offset address? 266
 Base    Offset  Old     New?
??????  000266  105764  ? JSR!537        ; Call patch space @ PATCH5
??????  000270  000016  ? PATCH+356
??????  000272  002002  ? BR!(Q&377)     ; Not a primary entry, go on
??????  000274  052715  ? ^Z
Offset address? 410
 Base    Offset  Old     New?
??????  000410  002044  ? NOP            ; Insure PPN table is always update
??????  000412  010502  ? ^Z
Offset address? ^Z
Base address? ^Z
Module name? ^Z
File to patch? ^Z
```

I'd like to thank Mark Hartman, the RSTS SIG Librarian, for descriptions of his troubles with the named directory code, and also for his NMEMGR program to manage NAME.SYS. If you are still not sure how to sysgen or use named directories, take a look at the Fall 1982 (Vol. 9 No. 2) RSTS SIG newsletter. (Yes, it's actually being printed again!) Mark has a nice article with instructions on how to sysgen and use named directories. Also, he tells me that his NMEMGR program should be on the Fall (Anahiem) RSTS SIG tape. A few notes on his article: 1) The disk name for UU.NME does in fact default to the system disk for lookups — just not for adds or deletes. 2) The NMEMGR INIT command does not create a space efficient NAME.SYS. The

# The way you write BASIC programs is about to change.

**Digital will be delivering Version 2 of BASIC-PLUS-2 and VAX-11 BASIC by March 1983. By this summer, you can improve the way you write BASIC programs by using the new BASIC data definitions and program control syntax. This major new release incorporates features that can enhance the readability and maintainability of your programs...thus decreasing software maintenance costs.**

Educational Services can help you take advantage of these new features. The BASIC, Version 2, Update Seminar offered for the first time in March 1983 will familiarize current users of VAX-11 BASIC and PDP-11 BASIC-PLUS-2 with all the enhancements of this major new release...and many more features that can help you build better BASIC programs...

*...from*

complex commercial or engineering applications

*...to*

development of "systems" programs

*...to*

the formatting of your source code.

Of course, the new release provides complete source compatibility with previous versions. So if you want to, you can keep coding BASIC the way you always have. But we bet you won't.

Enroll now! Fill out the Registration Form and mail it today. Enrollment is limited.

---

**BASIC, Version 2, Update Seminar**
**Registration Form**

_____ Register me in the following 2-day seminar. Registration fee of $600 is enclosed.

Authorized Signature _____ Title _____ Date _____

| | | | |
|---|---|---|---|
| _____ **Boston, MA** | March 28, 29 | _____ **Chicago, IL** | May 16, 17 |
| _____ **Dallas, TX** | April 4, 5 | _____ **Los Angeles, CA** | June 1, 2 |
| _____ **San Francisco, CA** | April 18, 19 | _____ **New York, NY** | June 13, 14 |
| _____ **Washington, DC** | May 2, 3 | | |

Full tuition of $600 payable to Digital Equipment Corporation must accompany registration.

_____ Please send me more information on this seminar.

Name _____ Title _____

Company/Institution _____ Division _____

Address _____ Telephone _____

City _____ State _____ ZIP _____

Mail to: Digital Equipment Corporation
**Seminar Programs**
12 Crosby Drive
Bedford, MA 01730

...Or call now to pre-register (617) 276-4949. Enrollment is limited.

CIRCLE 175 ON READER CARD

digital

recommended 64. block file is large enough for almost 1800 names, but NMEMGR writes the maximum count as 800. Ideally, the command should create the smallest file capable of holding a specified number of names. 3) NMEMGR has a minor bug in the WHO command — it forgets to tack the current device on to the string to be FSS'd. This is easily fixed. Again, thanks Mark!

## CONCLUSION

Next month, look out for some hidden mode bits in UU.MNT, the results of more testing of named directories, and how to get FMS FDV to display VT100 graphics.

If you aren't up to keying in these patches, send $20.00 to IISI (Attn:MCG) and we'll send you a tape of the patch command files, plus ONLRES, the load average stuff, and all the other goodies from the previous months. Hurry, though, because all of this stuff is starting to fill the small tape . . . Please specify 800 or 1600bpi.

I hope you have enjoyed this installment of the RSTS Crystal Ball. I will continue to try to present information which is interesting and useful. If you have any questions, gripes, or suggestions, call or write to me.

Until next time, JRST WIN!

Michael C. Greenspon
C/O Integral Information Systems
9832 Vicar Street, Suite 100
Los Angeles, California 90034
(213) 558-0732

♥



# EDITING SYSTEMS

David Spencer, Infinity Software Corporation

## ONLPAT Commands

One of the most frequently used and least documented programs that can be found on the RSTS/E distribution kit is ONLPAT. Anyone who has installed patches from either Digital or other software vendors is likely to have come in contact with it. In this article I will attempt to document as best as possible this wonderful system tool.

### 1.0 ORIGINS OF ONLPAT

The program we call ONLPAT is also something called INIPAT. INIPAT is the PATCH option found in the list of commands available from INIT.SYS when the system is "down." Like the disk INICLN "clean" code in INIT that became ONLCLN, ONLPAT is actually INIPAT with special I/O routines to allow it to operate in timesharing mode. Therefore all commands documented here should be identical to those in the INIT counterpart.

### 2.0 PURPOSE OF ONLPAT

ONLPAT is intended for use in patching SILs, Save Image Libraries. These are files like the RSTS/E monitor and other pieces of code like PIP that have been run though SILUS or MAKSIL.

However ONLPAT does not restrict you to using it on files with symbol tables. It is in fact capable of being used to modify any type of file.

(For those interested in symbol table layouts see either the MAKSIL source or Mike Mayfield's RSTS/E Monitor Internals manual.)

### 3.0 USING ONLPAT

During the SYSGEN process ONLPAT is copied to the system disk in account "[1,2]". The system build command files leave it there because it is intended to be used later by the automated patch facility and/or system managers who wish to enter patches from the Software Dispatch by hand before they receive their tape(s).

In this article I will discuss both the interactive and command file modes of ONLPAT.

First, let's look at a simple ONLPAT session and identify the various questions and options available.

```
RUN $ONLPAT
Command file name? <lf>
File to patch? <lf>
File found in account [0,1]
Module name? RSTS
Base address? ..CAGE
Offset address? 0
 Base   Offset  Old     New?
132544  000000  000010  ? 7      ; New cache age
132544  000002  103656  ? ^C
Patch complete

1 patch installed

Command file name? ^Z

Ready
```

### 4.0 COMMAND FILE NAME

The first thing ONLPAT asks for is a command file name. Entering a line feed or carriage return will instruct ONLPAT that you wish to use it interactively. (I will be discussing command files later on.)

### 5.0 FILE NAME

Either a filename or a line feed is acceptable. A line feed translates to the name of the currently installed monitor SIL. If the filename entered with a "/N" switch, this tells ONLPAT not to attempt to find the symbol table. Normally this switch is not necessary. ONLPAT looks to see if the file is in SIL format and disables symbolic patching if it is not. The only time you might want to use this switch is when you are attempting to patch a SIL and want direct access to locations in the file.

If no PPN is specified with the filename, ONLPAT checks the current account for the file. If it cannot be found there, it looks on account "[0,1]". Thus if you have a file with the same name in both your account and in "[0,1]", you must explictly enter the correct account number with your filename.

### 6.0 MODULE NAME

If the file you opened is a SIL and it also has multiple modules, then you can get a directory of the module names by either entering a "?" or a carriage return.

```
Command file name? <lf>
File to patch? <lf>
File found in account [0,1]
Module name? ?
Directory of SIL:
 Name    Ident     Load     Size  Transfer Total
RSTS    07.111  000000  133000  000001   23K
CRA     07.111  120000  002500  000001   24K
XVR     07.111  120000  006100  000001   25K
FMS     07.111  120000  010700  000001   28K
EMT     07.111  120000  015000  000001   31K
GEN     07.111  120000  013200  000001   34K
TER     07.111  120000  027200  000001   40K
DSK     07.111  120000  005200  000001   41K
FIP     07.111  140000  007000  000001   43K
OVR     07.111  002000  101000  000001   59K
DEFALT  07.111  001000  002000  000001   59K
Enter the name of the module in the SIL to be patched:
```

ONLPAT will accept only one of the module names listed in the directory. If a line feed is typed then ONLPAT uses the name of the first module in the directory.

### 7.0 BASE ADDRESS

For the base address one of two things is possible. First, a number may be entered for an absolute address into the module. The other option is to enter the name of a global symbol from the symbol table. ONLPAT finds the symbol and uses the value found with the symbol for the base address.

(Typically a global symbol references the starting address of a routine or a word of data. All through the monitor there are routines that are called and usually the name is the global symbol. Also in the monitor there are many words of data, such as the caching age shown in the first example, that can be modified to tune system performance.)

The base address may be entered with an optional argument delimited with a colon. The format is "x:y". If the file is a SIL then "x" and "y" can be either the module name and the overlay number, or the module number and the address of the overlay.

For those files that are not SILs, "x" is the block number minus one in the file, and "y" is the offset in the block. The reason "x" is the block minus one is that ONLPAT begins counting at zero and not one.

### 8.0 OFFSET ADDRESS

The offset address is much like the base address. It can be either a number or a symbol name. Line feed is also acceptable; if used ONLPAT starts with offset zero.

The offset address is added to the base address to shift to points inside a routine. For example if there were a patch to the routine "FOOBAR", an offset address of 100 might be used to get to the word of the incorrect code.

## 9.0 SIMPLE COMMANDS

This is an example of some of the simpler (but not to say that they aren't powerful nor useful) commands of ONLPAT.

```
>! Create a file to play with
>
>PIP FOO.BAR=KB:
FOO BAR BLETCH!
^Z
>
>! Now let's patch it
>
>RUN $ONLPAT
Command file name? <cr>
File to patch? FOO.BAR
Base address? 0
Offset address? 100
 Base   Offset  Old     New?
000000  000100  000000  ? 100  ; Patch in octal 100
000000  000102  000000  ? 100. ; Patch in decimal 100
000000  000104  000000  ? ^Z
Offset address? 0
 Base   Offset  Old     New?
000000  000000  047506  ? "WH  ; Change FO to WH
000000  000002  020117  ? <lf> ; Verify only
000000  000004  040502  ? ^    ; Go back to look
000000  000002  020117  ? <lf> ; Still there, move on
000000  000004  040502  ? <lf>
000000  000006  020122  ? <lf>
000000  000010  046102  ? <lf>
000000  000012  052105  ? <lf>
000000  000014  044103  ? <lf>
000000  000016  006441  ? <lf>
000000  000020  000012  ? <lf>
000000  000022  000000  ? '*   ; Put in a star
000000  000024  000000  ? ^C   ; Patch complete
Patch complete

1 patch installed

Command file name? ^Z

>! Look to see what we've done
>
>TYPE FOO.BAR
WHO BAR BLETCH!
*@d
>
```

A number of commands were exercised in this example. But first, a few concepts.

ONLPAT assumes all numbers are in octal. If you want decimal, follow it with a period ("."). Entering any number with an eight or nine in it without the period will return an error.

Working with ONLPAT is like using ODT. You can move up and down through locations and examine and deposit values. So any number that is typed into an opened location gets stored there. Depositing values into opened locations is one of the first things demonstrated in the example.

Next shown in the example was the use of control/Z or carret/Z. Both operate identically in ONLPAT. (In INIPAT, only control/Z is valid.) This command instructs ONLPAT to back off one command level or question. Thus it is possible to use control/Z to go backward all of the way out of ONLPAT.

The next command shown was the double quote. A double quote ("") followed by TWO characters converts the characters into their internal ASCII representations at that location. Notice how "FOO" became "WHO" this way.

Later on in the example a single quote is demonstrated. Much like a double quote, the single quote converts the following character and stores it in the opened location. Important note: a single quote creates the one byte of text and clears the other byte in the word.

By the way if you haven't noticed, comments can be entered by starting them with a semi-colon. This is not normally very useful for interactive patching, but does become important for use in command files.

Between the "WH" and the next command there were a number of lines not modified by using the line feed key. Line feed does not alter the current location but simply closes it and moves onward to the next one. This is used mostly for inspection and verification that patches were installed correctly.

The complement of the line feed is the carret ("↑"). It is used to go backward and open locations prior to the current one.

The last command displayed in the example was the use of carret/C. This is not a control/C. Typing a control/C to ONLPAT will abort the patch procedure and make no changes. A carret/C tells ONLPAT that the patch is finished. (For INIPAT, the INIT PATCH option, there is no difference, and either tells INIT that the patch is complete.)

There is one more command that exists but I didn't demonstrate it in the example. It is the percent sign ("%") command. As many as three characters can follow the percent sign and they are converted and stored in Radix-50 format.

## 10.0 MORE ADVANCED FEATURES

ONLPAT has many other features as well. The following example demonstrates these more advanced commands.

```
>RUN $ONLPAT
Command file name? <cr>
File to patch? FOO.BAR
Base address? 0
Offset address? <lf>
 Base   Offset  Old     New?
000000  000000  044127  ? \    ; Get into byte mode
000000  000000     127  ? <lf> ; Verify
000000  000001     110  ? \O   ; Octal display
000000  000001     110  ? \D   ; Decimal display
000000  000001      72. ? \'   ; Character display
000000  000001      'H  ? <lf>
000000  000002      'O  ? <lf>
000000  000003      '   ? <lf>
000000  000004      'B  ? <lf>
000000  000005      'A  ? <lf>
000000  000006      'R  ? <lf>
000000  000007      '   ? <lf>
000000  000010      'B  ? <lf>
000000  000011      'L  ? <lf>
000000  000012      'E  ? <lf>
000000  000013      'T  ? <lf>
000000  000014      'C  ? <lf>
000000  000015      'H  ? <lf>
000000  000016      '!  ? <lf>
000000  000017     015  ? <lf>
000000  000020     012  ? <lf>
000000  000021     000  ? <lf>
000000  000022      '*  ? <lf>
```

# YOU'RE NO SOFT TOUCH WHEN IT COMES TO SOFTWARE.

You're smart enough to buy only what you need—and we've unbundled all of MCBA® manufacturing software so you can afford everything you want!

- GUARANTEED PERFORMANCE
- ONE-YEAR UPDATE SERVICE
- THOUSANDS OF INSTALLATIONS NATIONWIDE

In addition, INSTALLATION, TRAINING, CUSTOMIZATION, EXCLUSIVE SOFTLINE™ TELEPHONE DIAGNOSTIC SUPPORT, and other services are available for a fee. Buy only as much additional support as you need.

**Manufacturing Software MCBA®**

**RSTS™ & RSX-11™ Versions Available Now!**

## CALL NOW 35% OFF!

- ACCOUNTING PACKAGES
- DISTRIBUTION PACKAGES
- MANUFACTURING PACKAGES
- PROFESSIONAL PACKAGES

CALIFORNIA SYSTEMS ASSOCIATES

2845 Mesa Verde Drive East, Suite Four,
Costa Mesa, California 92626
For demonstration call: (714) 546-9716

CIRCLE 163 ON READER CARD

```
000000  000023     000  ? <lf>
000000  000024     000  ? ^Z
Offset address? 0
 Base   Offset  Old     New?
000000  000000  044127  ? /D     ; Try decimal word mode
000000  000000   18519. ? /%     ; Get Rad-50
000000  000000    %KV9  ? /'     ; How about printable text
000000  000000     "WH  ? <lf>
000000  000002     "O   ? <lf>
000000  000004     "BA  ? <lf>
000000  000006     "R   ? ^Z
Offset address? 102
 Base   Offset  Old     New?
000000  000102  000144  ? @      ; Change base address
Offset address? 102
 Base   Offset  Old     New?
000144  000102  000000  ? ^Z
Offset address? ^Z
Base address? 0
Offset address? 102
 Base   Offset  Old     New?
000000  000102  000144  ? @@     ; Use location as offset
 Base   Offset  Old     New?
000000  000144  000000  ? @@100 ; Set offset manually
 Base   Offset  Old     New?
000000  000100  000100  ? ^Z
Offset address? ^Z
Base address? ^Z
File to patch? ^Z
Patch complete - no modifications requested
0 patches installed

Command file name? ^Z
>
```

The two things demonstrated here were the various flavors of byte and word mode, and the use of the "at" sign for indirection.

Byte mode has several modifiers that change the display for the byte locations. Entering a backslash ("\") and optionally an "O" for octal opens byte rather than word locations for display and possible modification. If you add a "D" after the backslash, the values of the byte locations are displayed in decimal instead of octal. If a single or double quote follows the backslash, locations that are printable ASCII are displayed in their character format.

Word mode is like byte mode except that, obviously, it works on entire words. The options for the backslash are the same as byte mode. In addition, there is the percent sign modifier. This performs the display in radix-50.

If a word mode command is entered on an even boundary while in byte mode, ONLPAT will switch back to word mode.

The "at" command has three flavors. The first way it can be used is by itself. This changes the base address to the value of the current location and prompts for the offset address. The second way the "at" sign can be used is by doubling it. This takes the value of the current location and uses that as the new offset. The last way the "at" sign may be used is by entering two and following them by a number. This number becomes the new offset address.

## 11.0 ONLPAT DOES MATH! (and other wonders)

Yes, ONLPAT can perform magic. The following are some brief examples of the neat things ONLPAT can do.

```
>RUN $ONLPAT
Command file name? <cr>
File to patch? FOO.BAR
Base address? 0
Offset address? 0
 Base   Offset  Old     New?
000000  000000  044127  ? 100=
Value =  000100, 64.
000000  000000  044127  ? 2+2=
Value =  000004, 4.
000000  000000  044127  ? 5-2=
Value =  000003, 3.
000000  000000  044127  ? 2*3=
Value =  000006, 6.
000000  000000  044127  ? 7/2=
Value =  000003, 3.
000000  000000  044127  ? 7\2=
Value =  000001, 1.
000000  000000  044127  ? 1!4=
Value =  000005, 5.
000000  000000  044127  ? 2&3=
Value =  000002, 2.
000000  000000  044127  ? 2#3=
Value =  000001, 1.
000000  000000  044127  ? 2^3=
Value =  000020, 16.
000000  000000  044127  ? ^C
Patch complete - no modifications requested
0 patches installed

Command file name?
>
```

The first thing you might have noticed is that you can get ONLPAT to print the value of expressions with a suffix of an equal sign. As shown above, ONLPAT gives the value in both octal and decimal.

Most of the math symbols are pretty obvious. A "+" is add, "-" is subtract, "*" is multiply, and "/" is divide. A backslash returns the remainder of divide. A "!" produces a logical "or" of the two numbers. A "&" returns the logical "and" of the two numbers. Using a "#" gives you a logical "xor"; otherwise known as exclusive or. The strangest one of all is the carret ("^") which performs an arithmetic shift (in MACRO-11 the ASH instruction). The first number is rotated to the left "n" times, "n" being the second number. (If the second value is negative then the shift will be to the right instead.)

## 12.0 ONLPAT UNDERSTANDS CONSTANTS

Besides being able to perform math, ONLPAT can also return values for constants and special variables.

```
>RUN $ONLPAT
Command file name? <cr>
File to patch? FOO.BAR
Base address? 0
Offset address? 10
 Base   Offset  Old     New?
000000  000010  046102  ? .=
Value =  000010, 8.
000000  000010  046102  ? Q=
Value =  046102, 19522.
000000  000010  046102  ? MOV=
Value =  010000, 4096.
000000  000010  046102  ? ADD=
Value =  060000, 24576.
000000  000010  046102  ? .FSS=
Value =  104064, 34868.
000000  000010  046102  ? .WRITE=
Value =  104004, 34820.
000000  000010  046102  ? ^C
Patch complete - no modifications requested
0 patches installed

Command file name? ^Z
>
```

ONLPAT has two special variables that it maintains. These are the dot variable (".") and the "Q" variable. Dot is equal to the sum of base address and the offset address. "Q" is equal to the value of the currently opened location.

ONLPAT has an internal database of values for PDP-11 instructions, and RSTS EMT's, UUO's, and various other RSTS specific things like FIRQB and XRB. Using this facility, patches can be created that look very much like MACRO-11 code. (Rifle through some back issues of the RSTS Professional. Persons like Michael C. Greenspon have a tendency to use this function of ONLPAT to its fullest.)

### 13.0 OTHER WEIRDNESS

ONLPAT has some more interesting numerical evaluations up its sleeve . . .

```
>RUN $ONLPAT
Command file name? <cr>
File to patch? FOO.BAR
Base address? 0
Offset address? <lf>
 Base   Offset  Old     New?
000000  000000  044127  ? 100<200=
Value =  000001, 1.
000000  000000  044127  ? 100>200=
Value =  000000, 0.
000000  000000  044127  ? 100<=200=
Value =  000001, 1.
000000  000000  044127  ? 100>=200=
Value =  000000, 0.
000000  000000  044127  ? 100<>200=
Value =  000001, 1.
000000  000000  044127  ? 100=200=
Value =  000000, 0.
000000  000000  044127  ? "FO?
Verification error
Patch complete - no modifications requested
0 patches installed

Command file name? ^Z
>
```

As you can see, ONLPAT can take two numbers and compare them against each other. If the comparison is true, then a one is returned. If the comparison is false, then a zero is returned.

The last line shows off the question mark command of ONLPAT to compare an expression against the open location. If the test is false, which is the case here since that word actually contains the text "WH", the patch is aborted. A very useful verification tool.

### 14.0 COMMAND FILES

At this point, explaining command files becomes not much more than a trivial task. This is because command files are essentially formatted logs of previous interactive patch sessions.

Consider the following command file to change the cache age of the RSTS monitor.

```
File to patch? <lf>
Module name? <lf>
Base address? ..CAGE
Offset address? 0
 Base   Offset  Old     New?
??????  000000  000007  ? 7     ; New cache age
??????  000002  ??????  ? ^C    ; Patch complete
```

That's all there is to it. Now watch what happens when this patch is applied with ONLPAT.

```
>RUN $ONLPAT
Command file name? EXAMPL.CMD
File to patch? <LF>
File found in account [0,1]
Module name? <LF>
Base address? ..CAGE
Offset address? 0
 Base   Offset  Old     New?
132544  000000  000007  ? 7
132544  000002  103656  ? ^C

Patch complete


1 patch installed

Command file name? ^Z
>
```

## 15.0 COMMAND FILE FORMAT

As said above, the command file looks like an interactive patch session. An easy way I've found to create command files is to run $ATPK, type the input, and edit the log later to create the command file.





The command file must look nearly exactly like an interactive session. The text of the questions must appear along with the responses. (The text may be in any mix of upper and lower case though.) Any line that contains a line feed must have as the input for that line the text "<LF>". All control characters are represented as carrets and the letter. The only external input possible is the name of the file to patch, which ONLPAT will ask for from the keyboard if neither a name nor the text "<LF>" appears.

## 16.0 QUESTION MARK USAGE

ONLPAT normally checks the base address, offset address, and contents of the locations to verify that they are the same. This can be selectively disabled by using question marks in those fields that might float or otherwise be different.

The following is a listing of a command file that will fail because some values will not match those in the monitor. Notice the use of question marks in place of values that might change and that they are indeed accepted.

```
File to patch? <LF>
Module name? <LF>
Base address? ..CAGE
Offset address? 0
 Base   Offset  Old     New?
??????  000000  000070  ? 7   ; The old must be 70
123456  000002  ??????  ? ^C  ; Look at base address
```

And now the session log . . .

```
>RUN $ONLPAT
Command file name? FAIL
File to patch? <LF>
File found in account [0,1]
Module name? <LF>
Base address? ..CAGE
Offset address? 0
 Base   Offset  Old     New?
132544  000000  000007  ? Old<>000070 7
132544  000002  103656  ? Base<>123456 ^C
Patch complete - no modifications made


0 patches installed
1 patch skipped

Command file name? ^Z
>
```

Comments are of course harmless in the text and very useful for explaining what is happening. As with MACRO code, I use and recommend them.

By far the best examples to read are the MONITR.CMD and other ONLPAT command files that appear on the patch kit tapes.

## 17.0 CONCLUSION

In closing I'd like to say that I hope that now some of the mystery of ONLPAT has been taken away from it perhaps more people will begin to appreciate ONLPAT and use it more often. I find it is an easy tool to use and a lot of fun to play with too. I hope you will find it so as well.

Many thanks to those who have read my work and responded. I wish you all many happy edits.

♥


# You can get more from your VAX computer or RSTS system with MAS-M.

MAS-M is the application software system from Martin Marietta Data Systems that can help you do more with your DEC hardware. That's because MAS-M is the on-line software system that gives you much more than you'd expect from packaged software.

## More Flexibility.

MAS-M's modular design lets you choose from 10 different application systems:

- Accounts Receivable
- Accounts Payable
- General Ledger
- Order Processing
- Invoicing
- Inventory Control
- Inventory Accounting
- Bill of Materials
- Material Requirements Planning
- Purchasing

You can implement just the modules you need to satisfy your demands. And no matter which combination you choose, the MAS-M system is always fully integrated.

MAS-M's flexible design also makes it easy to install, and simple for your users to operate. The MAS-M/VAX package is written in native mode VAX-11 BASIC and the MAS-M/PDP package is written in BASIC-PLUS-2 under RSTS/E. Both of these packages are based on the RMS data management system. These features make both MAS-M packages fully compatible with your current RSTS/E or VAX/VMS operating system.

## More Control.

You can count on MAS-M for more comprehensive data accuracy and security, too.

MAS-M's powerful transaction processing MONITOR gives you maximum control over your data—from start to finish. User passwords and menu selections are checked against user security profiles. Data entry validation is also standardized in the MAS-M MONITOR, so any invalid data can be corrected *before* it reaches your application program.

## More Productivity.

MONITOR is also an important tool in developing new applications. You can use MONITOR to create input screens and validation rules on-line. And, MONITOR can help you improve programmer productivity by providing a standard framework for input of code that minimizes the difficulties of user interface and terminal characteristics.

## More Support.

You can count on Martin Marietta Data Systems for system development and implementation, comprehensive training, and clear, concise documentation. We can also provide an extensive Maintenance Service to support your MAS-M system.

What it all adds up to is a packaged software system that can give you everything you need to get your jobs done. And more. Write or phone us today, and we'll tell you more about how the MAS-M software system can work for you.

# MAS-M

**The Software System That Can Help You Do More.**

MARTIN MARIETTA DATA SYSTEMS

Martin Marietta Data Systems
Marketing Services, R/H
6303 Ivy Lane, Greenbelt, MD 20770
(800) 638-7080 In Maryland (800) 492-7170

MARTIN MARIETTA

CIRCLE 111 ON READER CARD

# PRIVATE DELIMITERS

By David Patterson, Sivall's, Inc., Odessa, TX

With the release of RSTS V7.1 DEC gave us a new goodie called multiple private delimiters. These delimiters are local to a job, not a keyboard and are automatically cleared whenever the job enters a monitor wait (negative wait time). Being the hacker I am, I started playing with them as soon as I had a chance. The first thing I did was write a MACRO subroutine so that I could set and clear them from BASIC+2. During the debugging of this routine, I discovered that BP2's debug module can't handle the delimiters. This is not surprising since it was written some time before multiple private delimiters were set up. It is, however, rather frustrating so I came up with a patch for the user entry module that $DEBUG uses. While I was working on this, I discovered another problem, this one with the .SPEC directive to read the delimiters. It's actually just a documentation error. If no delimiters are set and a read subfunction is executed, an error 5 (NOSUCH) will be returned in byte zero of the FIRQB.

**DELIMI.MAC**

This is a BP2 callable subroutine that will set or clear a job's private delimiters. It has two entry points: SETDEL and CLRDEL.

SETDEL: This is the entry point to set the delimiters. It has two optional arguments, a string containing the characters to be used as delimiters and the channel to set them for. See the listing for details about the calls. The default delimiters are defined at label MASK; and currently consist of all characters except CTRL/S and CTRL/Q (this allows synchronization to work correctly). To change the default just alter the bit mask as required (see the system directives manual, .SPEC directive).

CLRDEL: This is the entry point to clear the delimiters. It has one optional argument, the channel number.

The default channel for both calls is zero.

**PAT000.MAC**

The module that is being patched is $STPDB. This module handles the user input for both $DEBUG and $STP (the stop thread), and who knows what else. Since we only want the patch to effect debugging, the first thing it does is to check for the presence of DEBUG. If it's not there everything continues as usual. If DEBUG is present, the patch saves the current delimiters, clears the delimiters, does the user input, and then restores the saved delimiters. This prevents DEBUG from trying to parse each character as a complete command.

What we have done at our installation is to put a patched version of the object module on LB: and to refer to it whenever we are debugging a program that uses private delimiters. For those of you who like to muck with the libraries, you can just replace the module in the BP2COM library but remember, DEC tends to frown on that. My apologies to those of you who use the BP2 resident library. We don't even have it on our system at present because we use RMS heavily and the 32KW limit won't allow the use of both RMS and the BP2 reslib. So, I haven't had an opportunity to play with it.

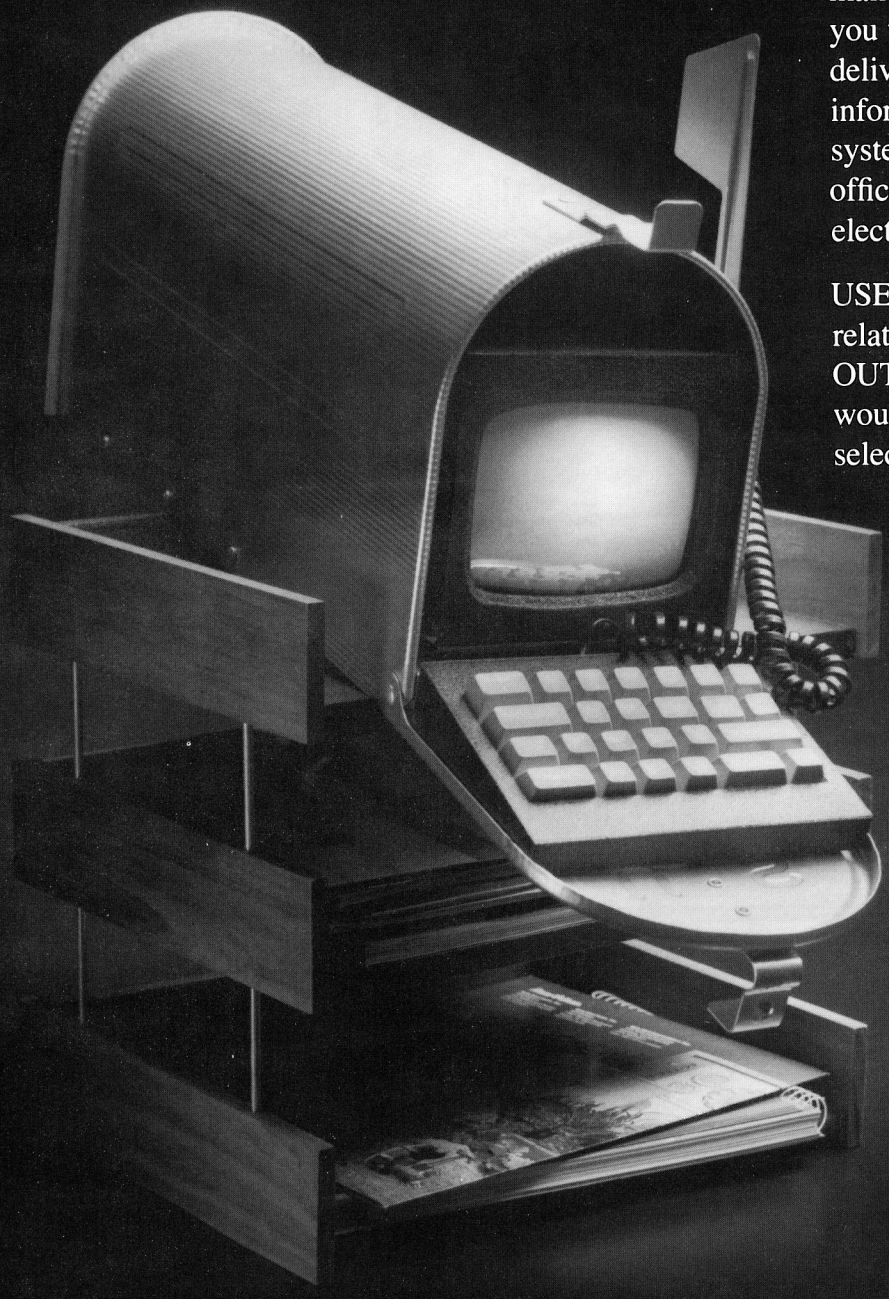
An example of installing and using the patched .OBJ file: (The checksums are valid)

```
MAC PAT000=PAT000                ; The patch file.
LBR TEMP=LB:BP2COM/EX:$STPDB     ; The needed module.
PAT STPDB=TEMP/CS:131101,PAT000/CS:53335 ; Patch it.
PIP LB:<40>=STPDB.OBJ            ; Put it where you can use it.

OLD EXAMPL
COMPILE/DEBUG
BUILD EXAMPL,LB:STPDB
TKB @EXAMPL
```

EXAMPL will now run with private delimiters and still allow you to debug it.

```
        .nlist  bin
        .nlist  bex
        .nlist  me
        .list   ttm
        .enabl  lc

        title   DELIMITER,<Private delimter subroutines>,01,11-Nov-82,<DMP>
        .sbttl  Comments and edit history.

;       Module name:            DELIMI
;       Date Written:           08-Sep-82
;       Author:                 David Patterson
;       Installation:           Sivalls, Inc.
;
;       Remarks:
;                 This module contains two entry points; SETDEL
;               and CLRDEL.  These two routines control the
;               multiple private delimiters for the user.
;                 These routines are callable only from BP2 at
;               this time and are called as follows.
;
;               CALL SETDEL         ! Set default delimiters on chn 0%.
;               CALL SETDEL(A$)     ! Set A$ as delimiters on chn 0%.
;               CALL SETDEL("",N%)  ! Set default delimiters on chn N%.
;               CALL SETDEL(A$,N%)  ! Set A$ as delimiters on chn N%.
;
;               CALL CLRDEL         ! Clear delimiters on channel 0%.
;               CALL CLRDEL(N%)     ! Clear delimiters on channel N%.
;
;       Linking instructions:
;
;               Compile this routine with MAC, (MAC DELIMI=COMMON,DELIMI).
;               Edit your ODL file to contain a reference to this routine
;                 (USER: .FCTR SY:filspec-DELIMI-LIBR).
;               or include it in the BUILD command (BUILD filspec,DELIMI).
;               Task build as usual.
;
;
;       Disclaimer:
;
;                The information in this document is subject to change without
;               notice and should not be construed as a commitment by either
;               the author or Sivalls, Inc.
;
;
;       Modificaton History:
;
;               Ver/Edit        Date            Modification
;
;                 00            08-Sep-82       Initial conception (DMP).
;                 01            11-Nov-82       Cleanup for release (DMP).
;


        .page
        .sbttl  Global symbols.

        .globl  setdel
        .globl  clrdel

        .page
        .sbttl  Code area.

        .psect  sisubs,rw,i,lcl,rel,con ; Sivalls private subroutines.
```

# ELECTRONIC MAIL. PRACTICALLY SPEAKING.



Sooner or later you will be using electronic mail. It just makes good sense. When you do, you will want a system that is complete—a delivery system, a scheduling system, and an information manager. Your electronic mail system will become an essential part of your office environment. USER-MAIL is such an electronic mail system*

USER-MAIL's power is easy to control. It relates to the way you work. Electronic IN, OUT, and HOLD baskets are just what you would expect. You can scan your IN basket, selecting only those message subjects you wish to read. Or, you can place a message into your HOLD basket for a number of days to have it automatically reappear in your IN basket on the appointed day. You can even have USER-MAIL recall specific messages by providing your own selection criteria. Replying, forwarding, and sending to groups are as easy as can be. And these are just a few of the features in store for you.

You owe yourself a closer look. Write for a brochure or give us a call direct.

NCCS USER 0110

**North County Computer Services, Inc.**
2235 Meyers Ave.,
Escondido, California 92025
(714) 745-6006, Telex: 182773

*USER-MAIL is currently available on DEC computers using the RSTS operating system.
RSTS is a registered trademark of Digital Equipment Corporation.
USER-MAIL is a trademark of Logic eXtension Resources.

CIRCLE 76 ON READER CARD

```
setdel: call    clrxrb                  ; The Set delimiter entry point.
        inc     xrb+14                  ; Function = 1 (set).
        mov     #mask,xrb+4             ; Default mask (all but ^Q & ^S).
        clr     r0                      ; Default channel is 0.
        tst     (r5)                    ; Any arguments?
        beq     dospec                  ; Nope.  Assume default on channel 0.
        mov     2(r5),r0                ; Yep get the string header.
        mov     (r0)+,r4                ; The starting address.
        mov     (r0),r3                 ; The length.
        beq     30$                     ; "" indicates default on channel n.
        sub     #40,sp                  ; Room for the mask.
        mov     sp,xrb+4                ; Address of mask.
        mov     sp,r0                   ; Clear the mask.
        mov     #20,r1                  ; 40 bytes long.
10$:    clr     (r0)+                   ;
        sob     r1,10$                  ;
20$:    clr     r0                      ; Build the mask.
        movb    (r4)+,r1                ; Get the next character.
        div     #10,r0                  ; Get the byte and bit offsets.
        movb    #1,r2                   ;
        ash     r1,r2                   ; Set the byte mask
        add     sp,r0                   ; Set the byte offset.
        bisb    r2,(r0)                 ; Set the bits in the mask.
        sob     r3,20$                  ; Do the next character.
        add     #40,sp                  ; Re-adjust the SP.
30$:    clr     r0                      ; Channel.
        dec     (r5)                    ; Is there a second argument?
        beq     dospec                  ; Nope.  Assume channel 0.
        movb    @4(r5),r0               ; Yep use it for channel.
        br      dospec                  ; And go do it.

clrdel: call    clrxrb                  ; The Clear delimiter entry point.
        clr     r0                      ;
        tst     (r5)                    ; Any arguments?
        beq     dospec                  ; Nope.  Assume channel 0.
        movb    @2(r5),r0               ; Yep.  Use it for the channel.

dospec: mov     #11,xrb                 ; Spec function.
        movb    #ttyhnd,xrb+7           ; Device handler (TTY).
        mov     #40,xrb+2               ; Byte count.
        asl     r0                      ; Channel times 2.
        movb    r0,xrb+6                ;
        .spec                           ; Do it.
        movb    firqb,r0                ; Any errors?
        beq     10$                     ; Nope.  Just exit.
        trap    377                     ; Let the BP2 error trap handle it.
10$:    return                          ; Back to the caller.

        .page
        .sbttl  Local subroutines.

clrxrb: mov     #xrb,r0                 ; Clear the XRB.
        mov     #xrbsiz/2,r1
10$:    clr     (r0)+
        sob     r1,10$
        return

        .page
        .sbttl  Data area.

;
;       Default delimiters (all but ctrl/S and ctrl/Q).
;         for details, see the systems directives manual V7.1
;         or later.
;
mask:   .byte   ^B11111111              ; 000 to 007.
        .byte   ^B11111111              ; 010 to 017.
        .byte   ^B11110101              ; 020 to 027.
        .byte   ^B11111111              ; 35 times (030 to 177).
        .nlist                          ;
        .rept   34                      ;
        .byte   ^B11111111              ;
        .endr;  34                      ;
        .list                           ;

        .end;   DELIMITER
```

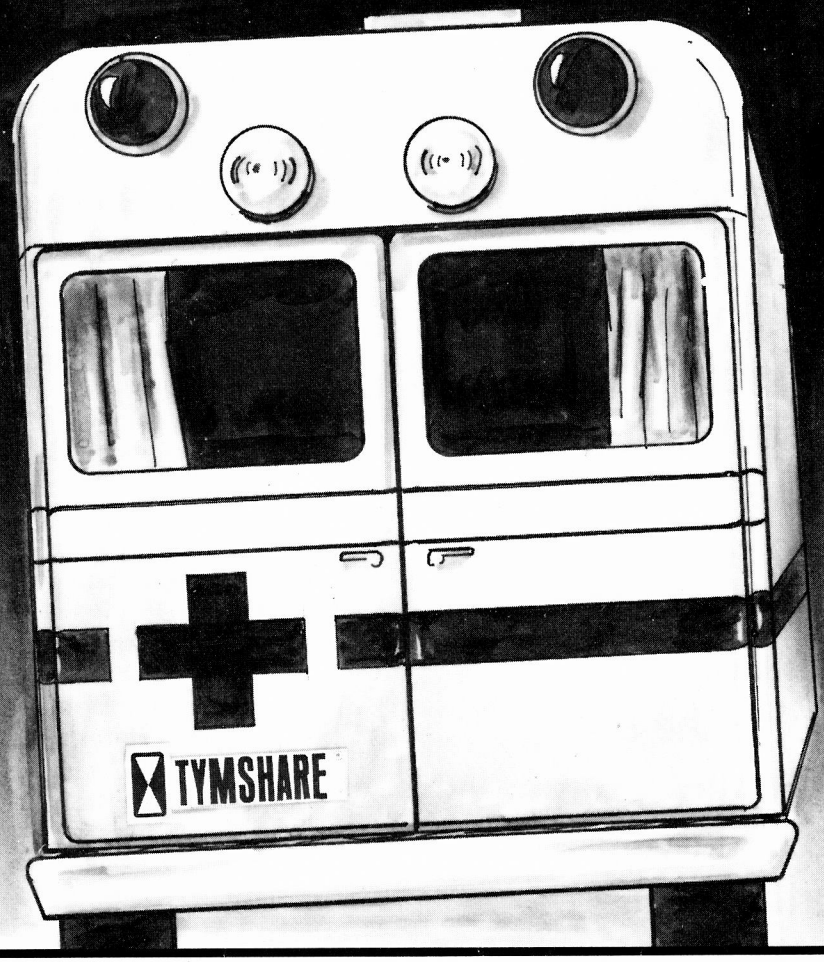
```
        .title  $STPDB                  ; PAT000.MAC
        .ident  /DMP000/
        .psect  BP20TS

;
;         Patch to allow DEBUG to work with a program that uses
;       multiple private delimiters.
;
;         The information in this document is subject to change without
;       notice and should not be construed as a commitment by either
;       the author or Sivalls, Inc.
;
        . = . + 346
lcl346:
        . = . + 56
$prmpt: mov     @#$otsv,r0
        tst     344(r0)                 ; Is debug present?
        beq     30$                     ; Nope.  Continue as usual.
        call    $clxrb                  ; Yep. Save the private delimiters.
        mov     #11,(r3)+               ; Private delimiters.
        mov     #40,(r3)+               ; 40(8) byte buffer.
        mov     sp,(r3)                 ; Put it on the stack.
        sub     #40,(r3)+               ;
        sub     #40,sp                  ; Adjust the stack pointer.
        tstb    (r3)+                   ;
        movb    #2,(r3)+                ; Handler = TTY.
        cmp     (r3)+,(r3)+             ;
        mov     #2,(r3)                 ; Function = read.
        emt     14                      ; Go for it.
        tstb    @#402                   ; Error?
        beq     10$                     ; Nope. Go save and clear them.
        add     #40,sp                  ; Yep. Assume it was 5 (NOSUCH),
        br      20$                     ;  fix the stack, and skip the clear.

10$:    clr     (r3)                    ; Function = clear.
        clr     -10(r3)                 ; Must be zero.
        emt     14                      ; Clear them.
        clr     @#402                   ; Assumes no error on clear.
20$:    mov     @#402,-(sp)             ; Save the 'Delimiters in use' flag.

30$:    call    ..rstt                  ; Assure column zero.
        mov     #20043,r3               ;
        call    $xwrt                   ; Display "#".
        tst     @#402                   ; Error?
        bne     lcl346                  ; Yes go handle it.
        call    $clxrb                  ; No. Prepare for the read.
        mov     @#$otsv,r0              ;
        mov     #400,(r3)+              ; 256. byte buffer.
        clr     (r3)+                   ; Must be zip.
        mov     26(r0),r1               ; Use the stash buffer.
        mov     r1,(r3)                 ; Address of Ibuff.
        emt     2                       ; Do it.
        mov     -(r3),r2                ; Number of bytes read.
        add     r2,r1                   ; Adjustment (unchanged).
        movb    @#402,r5                ; Save any error.
        tst     344(r0)                 ; Is debug present?
        beq     40$                     ; Nope.
        tst     (sp)+                   ; Were delimiters in use?
        bne     40$                     ; Nope.
        call    $clxrb                  ; Yep.  Restore them.
        mov     #11,(r3)+               ; Private delimiters.
        mov     #40,(r3)+               ; 40(8) byte buffer.
        mov     sp,(r3)+                ; On the stack.
        tstb    (r3)+                   ; Bump the pointer.
        movb    #2,(r3)+                ; Handler = TTY.
        cmp     (r3)+,(r3)+             ; Bump it again.
        inc     (r3)                    ; Function = write.
        emt     14                      ; Restore them.
        add     #40,sp                  ; Fixup the stack pointer.

40$:    tst     r5                      ; Was there an error on the read?
        bne     lcl346                  ; Yep. Let the standard code handle it
        return                          ; Nope. Return as usual.
        .end
```

## Our Australian Friends Know How To Exercise Mind And Body At The Same Time

A chess game in a Melbourne Park.

# Computer ills ???

# We'll Come To You




Rx When your computer system is down, you feel it. Tymshare's prompt emergency response will have your computer installation healthy in short order. To keep your system operating efficiently we will prescribe a custom preventive maintenance program which will insure maximum availability at the lowest possible cost.

If you're expanding your computer system, we have a full range of DEC and DEC compatible hardware — systems and subsystems. Our package includes installation and hardware support.

## COMPUTER SYSTEMS and SUPPORT DIVISION

TYMSHARE®

*For more information call...*
EAST: 10111 George Palmer Hwy./Bowie, Md. 20716/(301) 459-8363
CENTRAL: Bldg. 4/Suite 16/799 Roosevelt Rd./Glen Ellyn, Il. 60137/(312) 469-2600
SOUTHERN: 11999 Katy Freeway/Houston, Tx. 77079/(713) 870-0923
SOUTHWEST: 11050 Artesia Blvd./Cerritos, Ca. 90701/(213) 402-1114
WEST: 39100 Liberty St./Fremont, Ca. 94538/(415) 794-2490

NATIONAL HARDWARE SALES: 39100 Liberty St./Fremont, Ca. 94538/(415) 794-2538

* DEC and DIGITAL are registered trademarks of Digital Equipment Corporation

CIRCLE 36 ON READER CARD

# HOW TO SUSPEND A HIGH-PRIORITY CPU-BOUND JOB HOLDING YOUR SYSTEM HOSTAGE

By Greg Justice

This article is in response to Rick Powell's article in the Fall issue of the RSTS/E SIG newsletter, "The Cache Buffer." I have been in situations like those described in the article many times. However, most of the time crashing the program would do more damage to the software (data files, indices, etc.) than would letting the system die until it finished. Each of the situations was caused by an over-zealous system manager who liked to "hurry up" processing by raising priorities, especially on detached jobs!

I was running on an 11/34a with the pushbutton console and found that I could very easily get to the job control structure and suspend the job in about 5 of 6 cases. The exception is when the program gets hung in FIP and RSTS takes over the system. In this case crashing the program probably is the only course of action available short of crashing the system.

I am no longer running the 11/34a and have implemented the same process on an 11/70 with electronic console following the example in the article. The procedure is identical for steps 1 through 6:

1. Turn keyswitch to LOCAL.
2. Type Control/P on console.
3. System should respond with:
   CON =
4. Type an H to halt RSTS.
5. To which the system will respond:
   Hmmmmmmmm/Tnnnnn
   where mmmmmmmm is program counter (PC) address and nnnnn is a status register.
6. The second digit following the "T" indicates the CPU mode:
   4 = Kernel, 1 = User.
   You must be in User mode to proceed, if you get a 4 then go to step 11 and start over at step 4.
   Typical status register is T41410.
7. Now that we are halted in User mode we want to suspend the job.
   This is done by locating the job control structure of the job that was running when we halted. The job number is stored in a fixed location in low core (1006(8)). To check for a job type:
   1006/
   the system will type back:
   001006/xxxxxx
   where xxxxxx = the current job * 2 (in octal of course!
   if xxxxxx = 0 then the "null" job is running, so go to step 11 and start over at step 4.
8. Having determined that a "real" job is (was?) running we locate its Job Data block (JDB). The location of the current JDB is also stored in a fixed location in low core (1010(8)). To find the JDB type:
   1010/
   the system will type back:
   001010/xxxxxx
   where xxxxxx = the address of the current job's JDB.

9. Add 34(8) to xxxxxx to get the offset to the runburst/priority word in the JDB. Type the following to get to this word:

   yyyyyy/

   where yyyyyy is the result of adding 34 to the address above.

   the system will type back:

   yyyyyy/zzzzzz

   where zzzzzz = the runburst/priority word of the offending job.

   Some possible zzzzzz's are:

   003200 = -128 / 6 ! suspended job
   003370 = -8 / 6 ! normal job
   003000 = 0 / 6 ! slight boost
   003010 = +8 / 6 ! could be trouble

   The word is broken down with the runburst in the high byte and the priority in the low byte.

10. Type the desired runburst/priority word in the format shown in the list above. (000600 to suspend the job)

    The system should return to the CON= prompt at this point.

11. Type a C to continue the processor.

12. Type a Z to exit from the console emulator.

At this point your system should be "unhung" and the offending job can now be taken care of by more "normal" means (UTILTY, etc . . .).

I will not detail the procedure for the 11/34 because it follows this same method, the only difference is the procedure for examining and depositing addresses.

I have used the 11/70 procedure exactly as detailed and it does work. HOWEVER, all of this information, especially the fixed locations of the job structure, is subject to change by DEC.

The following is a sample of the procedure as I ran it.

```
^P                          !type control/P
CON= H00106764/T41410       !type H
CON= 1006/000024            !type 1006/
CON= 1010/011540            !type 1010/
CON= 011574/003174 000600 !type 011574/ then type 000600
CON= C R00000070            !type C
CON= Z                      !type Z
```

I would be interested in knowing if this procedure works on other 11's, or if anyone has come up with other solutions to this problem.

Most of the information in this article came from Rick Powell's article. Additional information concerning the job control structure was taken from Mike Mayfield's RSTS Internals Manual.

Greg Justice
Texas Distributors, Inc.
P.O. Box 344105
Dallas, TX 75234
(214) 620-1511

## Pros and Cons

# ABLE COMPUTER TECHNOLOGY ENABLE/34

By Allan Woloshin

About 18 months ago, we had a problem. We had a PDP 11/34 with 124KW of memory, one Able SUPER-MAX (a DH emulator), and two CDC 9762 disk drives, and we were trying to run 20 on-line jobs to do various accounting tasks. Each task varied between 12 and 31KW, with the average roughly 25KW. So we were constantly swapping out active jobs. The system ran very slowly; we regularly got complaints from our customers.

We tried everything. We knew that the problem was too much swapping, so we did everything we could to relieve the disk — we put most of our data files on one disk, and all of the swapping files on the other. We tried data caching, even though it was not recommended for small systems; it helped a little, but not enough. I also had a bad experiment with resident libraries: I tried to use RMSRES. The problem was, if RMSRES was resident, we could only have two and a half jobs resident. The two jobs using RMS would be 7K smaller, but RMSRES was 23K, for a net loss of 9K.

Finally, it came to buying more hardware. We had to do this cost effectively; we had very little cash (no pun intended). We quickly ruled out upgrading our computer, or buying a second computer (yet), which left us with only two choices: either add a swapping disk, or ENABLE.

The first choice was itself really two choices. Adding a swapping disk to RSTS does not necessarily mean adding a disk drive. There are memory devices on the market that have a controller which looks to RSTS exactly like a disk drive, but they run at memory-cycle speeds. Unfortunately, most of them are also "volatile," which means that when you shut off the power they erase themselves. (At that time, due to a lack of air conditioning, we had to shut down our computer every night and start it up again in the morning.)

ENABLE held the promise of adding more memory inline. This made a lot of sense to me; even at memory cycle speeds, it must take some certain amount of time to swap a job in and out. And besides, a swapping disk can only be read and written 512 bytes at a time, while memory can be in any increment desired. (If you want byte 10 from disk, you must read in 512 bytes.)

Of course we went with ENABLE, or I couldn't have done this story, could I? It might be important to know that we bought it through an OEM with our own C.E. doing the installation; we did NOT buy direct from ABLE.

Incidentally, there was a third option which we had not considered, since we did not know of it at the time. A PDP 11/34 can literally be transformed into a PDP 11/44 overnight. The PDP 11/34 CPU, FP-11 Floating Point Unit (if any), and DL11W are all removed, and replaced by one board: the PDP 11/44. This takes about 3 hours. Of course, the old PDP 11/34 memory must also be replaced, unlike ENABLE, which allows you to use it along with the new memory. Converting your PDP 11/34 into a PDP 11/44 also has the advantage of a much faster CPU.

Putting in ENABLE was more of a challenge than I had been prepared for, which is unfair, since the preliminary manual warned me of all that was to come. ENABLE allows you to use your old memory (18-bit address) in addition to the new (22-bit address) memory. The problem was, ENABLE plus the new memory had to go in its own backplane, near the end of the UNIBUS. Then the old memory has to go in its own backplane. Since the old memory was coming out of the first backplane on the bus (the CPU's backplane), we had to buy 2 more backplanes for the system. We had only bought one.

Well, we finally got the second backplane, and installed everything, and *Poof* RSTS booted. My C.E. said, "Well, there you go!" and left. I installed the ENACT patch, booted again and — WOW! We now had 252KW! Then I re-added XBUF and started up. It worked!

The next day, the problems started. My operator was doing backups at 5:00 AM (don't you just love 5:00 AM phone calls?). As soon as he mounted a magtape, the system hung. Well, we investigated, and discovered (after about 2 dozen CLEANs) that the tape drive always hung the system when ENABLE was active. However, SAV/RES worked just fine from INIT.

Then I remembered that ENABLE itself was a bus terminator; that the memory behind it was literally on a separate bus, created by ENABLE. So I removed the terminator card.

The tape drive now works perfectly. ABLE has narrowed this down to a "what" but not a "why." It seems that the standard bus terminator includes something called a B-SACK TURNAROUND. This is what was hanging the system. They sent me out [they loaned me] one of their own terminators, with the B-SACK TURNAROUND removed. The system worked fine (but they wanted their terminator back!).

Ever since then, I have met about a dozen people who told me that they "heard" that no one was able to install ENABLE — that many had tried, but few had succeeded. I can't say why.

Anyway, assuming you get ENABLE installed, it does exactly what it says it does. And you might be surprised to find out how much better a 252KW system runs than a 124KW system does! [The old, 18-bit address memory still removes the top 4KW, since it does not "know" that it should not anymore.] The relief was unbelievable — and we went from about five to ten complaint calls a day, to zero!

This same system is now running 640 KW of memory, supporting 35 jobs. And when needed, we can order more memory almost overnight, up to 2044KW.

We have not been entirely free from problems, however.

By far the biggest drawback is the reliance on ABLE Computers. No matter how sharp they are, how fast they work, we must always wait for them. Every time DEC re-releases RSTS, we must wait for the new patch. We got the patch for V7.1 only ONE WEEK before we received from DEC our V7.2 distribution kit! They claim that V7.0 to V7.1 was a big change and V7.1 to V7.2 is a minor one, the implication being that the V7.2 patch will get here much faster. We shall see — if it is soon enough, I will skip V7.1 altogether.

[I also intend to take a scratch disk, gen a V7.2 system, apply the V7.1 patch, and see how well it works. If I crash, no big deal.]

Another drawback is SPRs. As my system stands at this writing, I am still on V7.0. Every time a user task bombs with "?Odd address trap" or anything similar, rather than reporting an error to the job and continuing, RSTS crashes. Obviously I cannot report this to DEC; it must be the ENABLE patch.

Also, every time we cold-boot the system, during system startup (or during DEFAULT, if we do that first) we get the message

```
PARITY CSR
CONTROLS MORE
THAN 31K
FATAL RSTS/E
INITIALIZATION
ERROR!
OPTION:
```

When we try again, it works. Consulting the System Generation Manual, Appendix A, enlightens us with these words of wisdom:

Call the DIGITAL field service engineer.

Of course, the DIGITAL field service engineer wouldn't touch our system with a 10-foot memory cache. Listing the parity registers, we find that the message is correct — each CSR controls 128KW. So what?

I sent in an SPR to DEC. Their response: I had sent them incomplete information. What type of CPU was it: A PDP 11/23 or PDP 11/24?

The memory we have been using is National Semiconductor NS11L. When I explained this problem to ABLE, they asked me if that's what I was using. I was. They told me that they had heard of worse stories with that memory, and that it would be a good idea to stay away from it.

Now they tell me.

And finally, some problems I foresee but have not had yet: as our system grows, and supports more jobs, naturally we will add more on to it; more memory, more peripheral devices. And do more SYSGENs. You remember the small buffer problem that 11/70s had with V7.0? Remember how they fixed that with V7.1? There were two simultaneous solutions, but the more important one was to use data space.

Guess what: PDP 11/34s DON'T HAVE DATA SPACE! That means that when I start running low on small buffers, all I can do is weep. Of course, that is some way down the line. I have not yet hit the maximum SIL size on V7.0, and of course V7.1 will be even easier to work with. But it will come eventually, and probably unexpectedly and suddenly. And then all we'll be able to do is buy another computer.

Are the headaches worth the extra memory? I would have to say a hesitant: yes. But my advice to anyone considering buying ENABLE is: if you can afford some other way, such as putting a PDP 11/44 into your PDP 11/34 cabinet, by all means do it! ENABLE is only meant as a device for enlarging your current system, and NOT as a substitute for bigger systems!

**WEIGH FOR YOURSELF:**

| PRO | CON |
|---|---|
| comparatively low cost | difficult to install |
| ability to connect 4MB of memory | there are other ways to install memory |
| ability to use very large XBUF and data caching | most CE's have never heard of it; NONE carry spares |
| ability to load several resident libraries | must wait for patch before installing new systems |
| many jobs can be in memory | patch may conflict with DEC patches |
| | no DEC support |
| | CPU itself goes no faster |
| | when computer crashes, you don't know whether to call DEC or ABLE |

♥

# EVER MAKE A MISTEAK — Part 2

## or There's Even A Better Way

**(For Part 1 see RSTS Professional, June, 1982, Vol. 4, #3)**

By Paul A. Gilberti, Jr., Data Processing Manager,
Henschel Corp., Amesbury, MA

When the priority of a detached compute bound job is set to a value above all other jobs, the system will service only that job. To recover from this error, the system could be crashed and restarted; or all the users could wait until the detached compute bound job was completed. If the job went into a software loop, you waited forever. Crashing the system was also a very drastic solution. Not only did this waste the run-time the compute bound job had already received but it also made many other users very unhappy. Especially those who were updating files and had not yet closed their files.

Dr. James B. Wilkinson of the Erskine Mathematics Department provided a much better way of recovery in the June, '82 issue of the Pro. Halt the CPU, deposit an odd address in the program counter and continue. This causes the high priority compute bound job to bomb out with a "?Program lost-Sorry" fatal error. This solves the problem and restores the system to the users.

But what about all that run-time the compute bound job had already received and is now down the drain? Also what if the high priority compute bound job is also updating files and shouldn't be interrupted?

### With A Remote Diagnostic Console

Solution Two: Lower the priority of the compute bound job through the remote diagnostic console.

```
Step 1  Type !P                          ; Enter Console State.
Step 2  Type H                           ; Halt the cpu. This cause the system
                                         ;   to display the halt address and
                                         ;   status register.
Step 3  H????????/T?M???                 ; Examine status register being
                                         ;   displayed.
Step 4  If M = 4 then                    ; Make sure the processor
          Type 'Halt Address' LC         ; was halted in user mode,
          go to Step 2.                  ; otherwise try again.
Step 5  Type 1010L E                     ; Get the current job's JDB
Step 6  Type ######L                     ; Re-enter the location displayed.
Step 7  Type <lf> 14 times.              ; Move through the JDB until the
                                         ; priority is found.
Step 8  Type 003370 D                    ; Set priority/runburst to -8/6.
Step 9  Type the 'Halt Address'          ; Restart the CPU.
          followed by LCZ
```

### With A Switch Panel

Solution Two: Lower the priority of the compute bound job through the Switch Panel.

```
Step 1  Toggle the Halt                  ; Halt the cpu. This cause the system
                                         ;   to display the halt address.
Step 2  Record the 'Halt Address'
Step 3  Toggle 777776                    ; Examine status register being
        Load                             ;   displayed.
        Exam
Step 4  If status = 17xxxx or 14xxxx then ; Make sure the processor
          Load the 'Halt Address'        ; was halted in user mode,
        Cont                             ; otherwise try again.
        go to Step 2.
Step 5  Load 1010                        ; Get the current job's JDB
        Exam
Step 6  Add 34(8) to the JDB address
Step 7  Load the new address             ; Enter the location of the
                                         ; priority byte.
Step 8  Toggle 3370                      ; Set priority/runburst to -8/6.
        Deposit
Step 9  Load the 'Halt Address'          ; Restart the CPU.
        Cont
```

The high priority compute bound job is now running at a normal priority level, as is the rest of the system. ♥

# LETTERS to the RSTS Pro...

...continued from page 6

didn't work well, write in to tell us not to do it and why not!

Allan Woloshin
Data Processing Manager

*Dear Allan: Version 7.2 update is in the field and we will make every effort to release future updates of the RSTS Internals Manual as close as possible to the RSTS release date.*

* * *

For information to your readers I wish to point out an error which appears in the December 1982 article, *"Tips & Techniques — RSTS Job Context"*. The error occurs in the inadvertant duplication of one section of the sample program on page 52; second column starting with **".SBTTL CHECK THAT JOB STATUS . . ."** through the message **"?Program aborted"** on page 53 should be omitted. Other than this the sample program is reproduced properly.

I apologize for any inconvenience this may have caused readers. A machine-readable copy of the sample program can be obtained by sending a blank tape to me at the address below. Please specify the desired density (800/1600/6250).

Wef Fleischman, Systems Analyst
Software Techniques, Inc.
Los Alamitos, CA

* * *

I tried the Bill of Fare at the CHEZ RSTS, and I must say it was delectable. But I would like to advise Mr. DeMaria and your readers that a steady diet of CHEZ RSTS may result in a bad case of RSTS Runs.

My technical staff has advised me that the cure for this problem is $Pepto.Bis/No Dump.

We enjoy reading your magazine and derive great benefit from it. Keep up the good work.

A.R. A'Hearn, President
LogOn System, Knoxville, TN

* * *

RSTS Pro is my magazine!!!

Little bit of History — I worked on RSTS for four years as a programmer/analyst. Last March I had the opportunity to join the Royal Canadian Mounted Police. Since then I am very proud of that move, but, still, I haven't touched a RSTS terminal for 8 months. I didn't think I could make it, but so far with some reading of your magazine every week, I can live. If I find any good application for RSTS in my Work I will work on it to make it a good tool for us. Do you know any application which uses RSTS for Police Detachment? I would appreciate any information about it.

Thanks in advance . . .

Cst. Benoit Guay
RSTS Cop

* * *

We are looking for methods of reducing the tendency of all our disks to fill up, despite all the usual quota limitations, rude letters, threats and pleadings to which our users are becoming totally inured.

I want to tackle this by making it *really* easy for users to store files offline . . . an archive system, such as I have seen on a nearby DECSYSTEM-10. All the user does is type ARCHIVE FRED.DAT and the file disappears from his directory, into some system area for later transfer to disk or magtape to be held offline. He can get the file back, with a delay of half an hour (or half a day) by typing RETRIEVE FRED.DAT.

Does anyone know of such a system available for RSTS?

Geoff Draper
Computer Centre Manager
Australian Institute of Marine Science
Cape Ferguson, QLD

* * *

We have an 11/70 with 1.5 meg of memory running RSTS/E V7.0. All applications programs are written in house in DIBOL. I have about 15 users running the same inquiry program in response to telephone inquiries. I'm looking for a package that has multi-terminal support and will allow my staff to provide the 'guts' of the program in DIBOL. Goal: To have one program, one job slot, one set of buffers for open files for all 15 users.

Any suggestions and all calls are welcome. Enjoy your publication tremendously and find it extremely helpful.

Carol A. Edgar
Data Processing Manager
RM Electronics Co.
Grand Rapids, MI

* * *

I was reading your Letters to the Editors column and noticed a letter by Greg Steinkuhler about TECO and VTEDIT. Greg ended his letter with the wish that somebody would rewrite VTEDIT to be key compatible with KED and EDT. If Greg reads the September 1982 issue of the RSTS Professional, he will discover that I had already done so.

The VTEDIT.TEC I wrote needs some more work, but is very fast and runs in only 5K. Anyone who is interested in upgrading it to a complete emulation of KED/EDT has my cooperation and blessings.

I would like to apologize for typeset Macro-11 code in my article from the last (December 1982) issue. I was able to prevent the printers from doing justification on the code. I hope that no one has had a hard time in reading it.

I would also like to apologize for the letter I sent in the last issue on the 7.1 release of EDT. Time was short so I had to hastily type it at home on my typewriter. Apparently the publishers were also pressed for time.

David Spencer
Infinity Software Corp.
Santa Monica, CA

* * *

I have just received Vol. 4, #4 of *RSTS Professional* [Aug. 1982] — boy, you have no idea how relieved I was! You blokes probably think two months isn't NEARLY enough time between issues, but over here, I was convinced my subscription had run out and the renewal notice had been shredded by the mail sorting machine and I was doomed to a life without *RSTS Pro...!* Anyway, I couldn't let this issue pass without replying to Carl's observations about "home computers" [p. 4].

I have been a fan of DEC computers (particularly PDP/8 with 338 Programmable Buffered Display and a certain "War Games" program) ever since 1967. When I toggled my first machine language program into that PDP/8-I using the SWITCH REGISTER (oh happy days) I was HOOKED!! Since then, at various times, I have worked with a PDP/8e using OS/8 and PAL-8, COS-310 systems using DIBOL-8, a PDP11/34 using RT-11, a CTS-300 system using DIBOL-11, a VAX-11/750 system using VMS and (yech) COBOL and, of course, a PDP11/70 with RSTS!!

— and I've loved them all (except, perhaps, the COBOL); but I wouldn't have them in the house... (well, I might make an exception for the aforementioned PDP8 with display, etc. if it very tame). Which is what I'm writing about. Carl asks, amongst other things, "What is a 'home computer'?"; well, for my part, in a world with Prestel, The Source, packet switched networks, Medlars and various other "public access" data bases, a "home computer" DOES NOT NEED 256K of memory, OR 10Mb of disk, OR an "operating system" with Multi-terminal, Multi-user, Multi-programming, Multi-tasking capabilities! It's LUDICROUS!! What with you trying to put RSTS systems in the Lounge room and all the "micro" manufacturers trying to put "all-singing, all-dancing" UNIX "look-alike" or PICK systems in the study, the "home computer" is in danger of sprouting more terminals than the average IBM system has disk drives (perish the thought!).

Assume, for the moment, that we don't want to play games with our "home computer" — well, not much, anyway. For somewhere around $1000 you can get a 48K micro (I don't care what brand, there's plenty around) with a reasonable keyboard, 64 or 80 column screen, 150K mini-floppy, an RS232 port and a parallel port. Another $300 should get you a direct-connect modem with telephone connection or an acoustic coupler. For a family of four that adds up to $5200 to get everyone doing his own thing AS WELL AS talking to the outside world. The new Tandon Winchesters will soon be appearing in such things as the Corvus "Constellation" which should then allow you, if you're that depraved, to add a shared hard disk and printer to your "home computers" to form a LOCAL AREA NETWORK (albeit small) for around an extra $7000. So your family computer system is a full "four terminal multi-user" system for roughly $12,500!! If you add three terminals (VT100's, of course) to the MICRO-11 I suspect you will be up for about $14,000 WITHOUT each terminal having the ability to talk separately to the outside world.

I LOVE RSTS!!... Please don't get me wrong! I just don't think such things belong in the home!!

Finally, let me take issue with some of the questions you posed in your second last paragraph:

(a) What's so friendly about "No Logins"?

(b) Yes, they DO have "RECORD I/O" (that's all they have, usually);

(c) MDBS I and MDBS III ( and variants thereof) can now be obtained for most Zilog and Intel processors and it eats RMS for morning tea!

(d) TECO (as far as I remember) wasn't invented by DEC; they just started supporting it when they saw how good it was — if the blokes who wrote it in the first place ever cobble up a version for the 300,000 (at least) 8-bit micros in the world, they'll probably make a SECOND (?) fortune (!?); (N.B. I think EVERY computer in the WORLD should have at least TECO — if it can't have EDT!)

(e) in the two years I worked with RSTS recently, I never even SAW "GRIPE", let alone used it;

(f) with a "Constellation" or "Acorn" type Local Area Network who needs BATCH and SPOOL anyway — they only take up valuable Job Slots;

(g) utilities such as BACKUP and SAVER come as standard with such things as the "Constellation" and MDBS;

(h) if your system doesn't spend its time fragmenting its disk space (and it only has to service one user) who needs REORDR?

(i) most micro systems include ALL the bumf about their disk organization because their "DOS"s are so slack they EXPECT you to rewrite them;

(j) SYSTAT is more or less useless if you have only 1 task, 1 terminal, 1 program and 1 user;

(k) any VT5xxx program is pointless without a VT5y terminal;

(l) LOCK-11 doesn't come with RSTS;

(m) never, but neither was the only TRS-80 and Apple 11 I ever worked with;

(n) NO! (thank goodness!);

(o) YES

Now let me ask YOU some questions. Did your $10,000 include a RSTS license? AND 4 terminals? AND a GOOD dot-matrix printer (with decent underlines and lower case descenders)? AND 4 direct-connect modems for talking to the world at large? AND the infamous RMS?

I repeat — I am a DEC machine fan(atic) and I LOVE RSTS — but I still wouldn't keep a full-fledged system at home!!

Thank you all for a magnificient magazine — I would love to contribute but I'm still a "babe in arms" compared to most of your regulars. I read every issue avidly (and I'm STILL waiting for the "full color Dungeon map" ...)

Noel Goddard, AUSTRALIA

*Dear Noel: While I realize that a "home" RSTS System is not for everyone, the idea that you can configure a RSTS system (including a license) for under $10,000 puts it directly in competition with the genuine "home" computer.*

*I have noticed that a major use of the home computer is communicating with the outside world. I have been doing this for the past ten years with a terminal and modem quite adequately. It seems a bit of overkill to use a computer to talk to a computer; why emulate a terminal with a computer when a terminal would do the job more efficiently.*

*While a single terminal RSTS system may only marginally compete with the micro world, doubling or quadrupling the number of terminals yields a four user system that can do real computing for a whole family at once. And when a new family member comes along, you simply add a terminal.*

*Software on the micros is still primitive. Utilities have no standards and are supported by companies that come and go weekly. RSTS software offers a wide varity of mature application programs and system utilities.*

*A "home" RSTS System is only one choice among many possibilities. That's what makes horse races . . . $2 to win on RSTS please.*

*P.S. Hope you enjoy the dungeon map.*

* * *

In a letter to the RSTS Pro (Vol. 4, No. 6, December 1982) a reader reported a "possible . . . RMS SYstem Bug. It appears that if two files are opened on the same channel (even if the first one is closed before the second OPEN statement is executed), then BASIC+2 under RMS will expect the second file to be located on the same device as the first file, if there is no explicit device specified for the second file. It does not default to the system disk."

What your reader is seeing is actually a difference between RMS-11 and non-RMS handling of device name defaults. BASIC-PLUS, and BASIC-PLUS-2 not using RMS-11, behave differently from all languages using RMS-11. In fact, this is how RMS-11 works on all operating systems (RSTS/E, RSX-11M and M-PLUS, IAS, and Professional Operating System). When RMS-11 is not given a device name, it defaults to whichever device was last assigned to the logical unit (LUN, or channel). This behavior is compatible with the FCS-11 handling of LUNS, and the reason for it (which may now be a bit dated, given the newer logical naming facilities available on some systems) is to allow programs to specify a channel (rather than an explicit device) which allows the binding to a specific device to be deferred until task-build (or, in RSX environments, task installation) time.

This had been the behavior since Version 1.0 of RMS-11 and BASIC-PLUS-2.

To ensure that the system disk or public disk structure is used, SY: should always be specified in the file specification. This works in BASIC-PLUS, BASIC-PLUS-2 without RMS-11, and all languages using RMS-11.

Simon Szeto

* * *

I thought you might like to see this newspaper clipping regarding Teco.

# Freeze-Dried Dog Must Stay Home

**Columbus, Ohio**

**Franklin County Municipal Court Judge Donna Bowman says she does not want a freeze-dried dog admitted as evidence in a suit against a veterinarian.**

**Isabel Burks of Columbus has charged Dr. Dilbagh S. Kooner with malpractice in a lawsuit over the death of her dog, Teco, on April 13, 1980.**

**Burks had Teco freeze-dried after its death and now keeps the body in a terrarium in her home. She had planned to submit it as evidence in the trial, set for January 4, but Bowman said she might ask that pictures be used instead. The judge noted that the court stenographer would be responsible for the dog if it becomes an exhibit, which could mean keeping the dog for 30 days.**

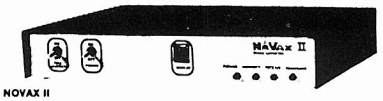
**The lawsuit seeks $771.18 in actual damages and $5000 in punitive damages.**

**Burks claims that Kooner misdiagnosed the animal's illness and says surgery could have saved its life. Kooner says he did not have time to make a proper diagnosis because Burks would not leave the dog overnight.**

***United Press***

Maybe this doesn't answer the questions: "How TECO?" and "Why TECO?"

Also, care to guess what the "NOvax II" is? Could it be something for the 11/70 user who doesn't really want to upgrade to a VAX? I'll trade you the answer for a T'shirt.




**NOW TWO MODELS TO SERVE YOU BETTER**

Paul E. Anderson, Vice President
Gallo Sales Company, Inc.
South San Francisco, CA

*Your T-shirt is on its way . . . what's the answer?* ♥

**AUTHORS!!!** Send your articles of interest to the RSTS community to the ***RSTS Professional*** on mag tape in either RNO, PIP or WORD-11 format. Eighty percent of this issue was transmitted via tele-communications from author's mag tapes to phototype-setting equipment and was not retyped.

# A GOLDEN SECTION SEARCH

By R. Frazer, Applications Analyst, On-Track Systems

Recently a client requested an extension of our amortization program (see !AMORT.BAS, RSTS PROFESSIONAL Volume 3,#2,page 57); he would like the ability to find an interest rate when given the opening principal, term, and term payment. In looking at my only formula, which solves for monthly payment, I could not see an easy way of solving for this variable (it's also been a long time since Algebra II).

The easiest way (and the most fun) to get the man his merchandise is to send the monthly payment formula a succession of interest rates until the solution approaches the given. The binary search cuts the domain subset in half every iteration, whereas the golden section finds two midpoints, zeroing in on 40%, or possibly 22%, of the same subset.

The "natural" illustration of this irrational number G is the fact that of two midpoints, the ratio of the first to the second is the same as the ratio of the second to the whole, or 1-G:1 = G:1-G


```
1       EXTEND
10      !       INTFIG.BAS      R.F., from AMORT.BAS                    &
        !                                                               &
        !       GIVEN TERM, PAYMENT, AND BALANCE, FIND INTEREST RATE    &
        !
60        Z = .1E39                     ! DEMAND SCALE FACTOR ZERO
100       INPUT 'PRINCIPAL BALANCE '; I.B                               &
        \ GOTO 32767    IF I.B = 0.                                     &
        \ INPUT 'NUMBER OF PAYMENTS'; I.N%                              &
        \ INPUT 'MONTHLY PAYMENT   '; I.M                               &
        \ PRINT 'ANNUAL INTEREST RATE = '; FNSEARCH(I.M,0.01,99.)       &
        \ PRINT                                                         &
        \ GOTO 100                                                      &
        !
1100    DEF FNX(X) = FNM.PMT(X)                                         &
        !
1200    !       MONTHLY PAYMENT CALC                                    &
        !                                                               &
        DEF FNM.PMT(A.I)                                                &
        \ I = A.I/1200.                 ! PERIOD INTEREST PERCENT       &
        \ X = 1. + I                                                    &
        \ X = (X^I.N%)/(X^I.N% - 1.)    ! FACTOR FOR FORMULA:           &
        !                                                               &
        !       M = B * I * (1 + I)^N / ((1 + I)^N - 1)                 &
        !                                                               &
        !       WHERE:  M = MONTHLY PAYMENT                             &
        !               B = PRINCIPAL BALANCE TO AMORTIZE               &
        !               I = PERIOD INTEREST RATE                        &
        !               N = TERM (NUMBER OF PERIODS)                    &
        !                                                               &
        \ Z = I.B * I * X                                               &
        \ M = FNROUND.2(Z)                      ! MONTHLY PMT           &
        \ M = M + .01   IF M < Z                ! ROUND UP              &
        \ FNM.PMT = M                                                   &
        \ FNEND                                                         &
        !
5000    DEF FNGOLD%(X,Y)                ! GOLDEN SECTION POINTS         &
        \ Z = (Y - X) * 0.381966                                        &
        \ G1= X + Z                                                     &
        \ G2= Y - Z                                                     &
        \ FNGOLD% = (Z = 0.)                                            &
        \ FNEND                                                         &
        !
5010    !> FNSEARCH      USE GOLDEN SECTIONS TO FIND CLOSEST            &
        !               APPROXIMATION; CREATE YOUR OWN FNX() AS USED    &
        !               IN LINES 5012, 5016, WHERE FNX() SLOPE FROM D1  &
        !               TO D2 NEVER REACHES ZERO (PEAK OR VALLEY)       &
        !                                                               &
          DEF FNSEARCH(Z,D1,D2)                                         &
        \ Z1 = ABS(Z)                                                   &
        \ T1, G2 = D2           ! INITIAL DOMAIN MAX                    &
        \ B1, G1 = D1           ! INITIAL DOMAIN MIN                    &
        !
5012      Q = FNX(G2) - Z1              ! TRY HIGH MIDPOINT             &
        \ IF  ABS(Q) < 0.0001                                           &
          THEN  Q = G2                                                  &
        \       GOTO 5022               ! CLOSE ENOUGH                  &
        !
5014      IF    Q < 0.          ! F(G2) < SEARCH VALUE Z1               &
          THEN  B1 = G2         ! NEW BOTTOM                            &
        \       GOTO 5020                                               &
        !
5016      Q = FNX(G1) - Z1              ! TRY LOW MIDPOINT              &
        \ IF    ABS(Q) < 0.0001                                         &
          THEN  Q = G1                                                  &
        \       GOTO 5022               ! CLOSE ENOUGH                  &
        !
5018      IF    Q > 0.          ! F(G1) > Z1                            &
          THEN  T1 = G1         ! NEW TOP                               &
          ELSE  T1 = G2         ! BETWEEN MIDPOINTS                     &
        \       B1 = G1                                                 &
        !
5020      GOTO 5012     UNLESS FNGOLD%(B1,T1)                           &
        \ Q = -1.                                                       &
        !
5022      FNSEARCH = Q                                                  &
        \ FNEND                                                         &
        !
20006   !>      FNROUND.2  ROUND FLOATING POINT NUMBERS TO 2 PLACES     &
        !                                                               &
          DEF FNROUND.2(Z) = FIX(100. * Z + .5) / 100.                  &
        !
32767     END                                                           ♥
```

# EMULEX TALKS DEC 2

## RADICAL RADIAL...

*True, the radial hookup scheme of DEC's UDA-50 allows you to drop a drive without saying "good-bye" to your entire system. But, is this really an advantage with new drives boasting long MTBF specs. Emulex controllers let you daisy-chain your drive connections using fewer, shorter (and cheaper) cables.*

## SEEK AND YE SHALL FIND...

*The UDA-50's ability to stack 16 seek commands does boost throughput—mainly for single drive systems. For all you multi-drivers, however, speedup isn't as pronounced. An Emulex-controlled multi-drive system stacks its seek commands (in effect) via its built-in system of overlapped seeks. Plus, overlapped seek and search commands (new to DEC in the UDA-50) already operate in Emulex controllers under all DEC operating systems.*

## TO ERR IS HUMAN...

*The 80-bit ECC of the UDA-50 can catch a lot of errors—it has to: High bit densities (try 11.4K bits per inch) on state-of-the-art media make 80-bit error correction a necessity, not a feature. And, the trade-off for correcting all those densely packed bits is loss of performance in skipping rotations every time an error occurs—All this in contrast to Emulex's proven 32-bit ECC.*

## PUTTING ON THE BRAKES...

*To slow the 2 MByte transfer rate of the disk to 800 KBytes at the Unibus, the UDA-50 uses a hefty 12 sector buffer. This means the UDA-50 can transfer 16-19 contiguous sectors at most before it skips a rotation and makes your software cry, "Uncle!"*

*In almost all applications, Emulex controllers can handle full (repeat full) track transfers of contiguous sectors and spiral read/write across cylinder head boundaries—and never skip a rotation. Why? Emulex passes data to your memory at rates much closer to those coming off your drives.*

## THINGS YOUR MOTHER NEVER TOLD YOU...

*For a complete report on these and other UDA-50 matters, write to Emulex.*

## FROM THE EMULEX FILE...

*Results for the First Quarter, Fiscal Year 1983 are in: Revenues up 100 percent, net earnings up 109 percent, earnings per share up 100 percent (all compared to the same quarter last year). Check your latest Emulex mailing for price reductions on some Q-bus and Unibus products. Not on our mailing list? Write: Emulex Corporation, 3545 Harbor Blvd., P.O. Box 6725, Costa Mesa, CA 92626. Or better yet, telephone us toll free at (800) 854-7112. In California, that's (714) 662-5600, and let's talk DEC.*

EMULEX

DISK • TAPE • COMMUNICATIONS



CIRCLE 58 ON READER CARD

# MLTJOB.BAS

By Alton O. Moore, McAllen, TX

Some time ago, when terminals were more plentiful around here, I had the privilege of having two on my desk. Though having two terminals upset the purist in me, it was certainly convenient. If I was interrupted in the middle of an edit and needed to use a terminal, I just turned around and there was another. I saved a lot of system resources which entering and exiting EDT would have used, because interruptions are plentiful around here.

Nothing good lasts forever though, and the next terminal shortage gobbled up my extra terminal. Being of stubborn mind, I mused about how I would fill this gap. After all, the extra terminal had turned out to be quite useful. I tried getting GSYS, a split-screen program from an old **RSTS Professional**, to work, but it had a bug which none of us wanted to chase. ATPK seemed interesting because it could control a job, but I wanted to control two jobs, not just one. Also, I had to be able to edit (of course) with the controlled jobs.

I investigated the use of pseudo keyboards, and MLTJOB (Multi-Job) is the result. MLTJOB allows a privileged user to start two jobs and do normal processing or programming on them. If a toggling accident should occur (if the controlling job is killed, for example) they are detached. If there is a disadvantage to using MLTJOB, it is the CPU time involved in transferring data from the pseudo keyboards' output buffers to the screen. This can be quite significant. On our system, doing a full SYSTAT under MLTJOB will run up 1 second of CPU time for the controlled job (doing the SYSTAT) and 1 second of CPU time for the job running MLTJOB. This has not turned out to be a problem, however, since we are heavily disk-bound. Your system might not appreciate it, though.

This version of MLTJOB has been in use for at least a couple of weeks with no program bugs detected. Heed the warning in the front pages of this magazine, however; this program is by no means a finished, polished, perfect version. Had I waited until MLTJOB was perfect before submitting it, it would never have reached print. This version is provided as a working model which will give you some insight into how pseudo keyboards work and how to use them. The program has in it some code which tailors it to work nicely on our system, such as the printing of a heading when you change to a job which is in command state (CCL command X). Also, our editor initialization files include the definition of CTRL/R as "refresh the screen" so that switching to a job running EDT will automatically put you right back where you were. What could be nicer?

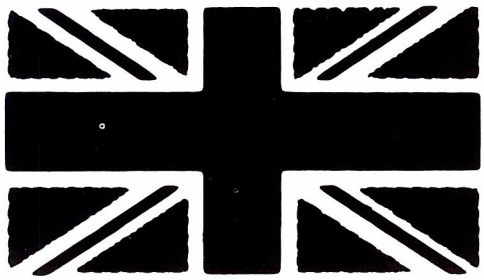
If you have any correspondence concerning MLTJOB.BAS, you may address it to: Jones and Jones, 2100 S. 10Th St., McAllen, TX 78501, Attn: Alton O. Moore

P.S. Watch for the MACRO-11 version, coming soon!

```
1 EXTEND
!************************************************************!
!                                                            !
!       Author:         Alton O. Moore, III                  !
!                       Data Processing                      !
!                       Jones & Jones, Inc.                  !
!                       McAllen, TX. 78501                   !
!                                                            !
!       System:         PDP 11/70 running RSTS/E version 7.1 !
!                                                            !
!       Program:        MLTJOB.BAS                           !
!                                                            !
!       Function:       MLTJOB allows any privileged RSTS/E  !
!                       user to control two jobs, much like  !
!                       some versions of DEC's OPSER. It will!
!                       pass data between the controlling and!
!                       controlled jobs in binary form, thus !
!                       allowing it to be transparent to     !
!                       screen editors.                      !
!                                                            !
!                       Typing two consecutive nulls on the  !
!                       keyboard will cause MLTJOB to toggle !
!                       from one job to another.  Sending one!
!                       null followed by ^\ will cause MLTJOB!
!                       to attempt to terminate.  If either of!
!                       the two controlled jobs is not at ^C !
!                       state (command level), MLTJOB will   !
!                       beep and refuse to end.              !
!                                                            !
!                       MLTJOB can be either run or installed!
!                       as a CCL command.  If installed as a !
!                       CCL command, MLTJOB/300 will cause   !
!                       the number of characters buffered by !
!                       the system from the controlled jobs  !
!                       to change from the maximum, 128, to  !
!                       10, for ease of use on slower keyboards!
!                       (i.e. 300 baud keyboards).           !
!                                                            !
!                       MLTJOB is immune to CTRL/C; if det-  !
!                       ached, however, binary input mode will!
!                       be disabled, and MLTJOB will be      !
!                       disabled when reattached to.         !
!                                                            !
!************************************************************!
!&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

5 CORE$=SYS(CHR$(7%)) \PK.GET.COUNT%=128                          &
        \PK.GET.COUNT%=10% IF CORE$="MLTJOB/300"
10 OPEN "KB:" AS FILE #10%, MODE 1%+32%                           &
        \FIELD #10%, 128% AS KB.BUFFER$                           &
        \! Open up the console KB: in binary mode.

15 LSET KB.BUFFER$ = CHR$(13%)+CHR$(10%)+CHR$(10%)                &
                +"Setting up multiple jobs; please wait..."       &
        \PUT #10%, COUNT 43%


! Open up the next 2 available pseudo keyboards and log then in..

20 LAST.KB.NO.TRIED% = 1%
22 LAST.KB.NO.TRIED% = LAST.KB.NO.TRIED% + 1                      &
        \PKB$ = "PK" + NUM1$(LAST.KB.NO.TRIED%) + ":"             &
        \ON ERROR GOTO 30
26 OPEN PKB$ AS FILE #1%, MODE 1%                                 &
        \GOTO 32
30 RESUME 22 IF ERL = 26 AND ERR = 8                              &
        \PRINT "NOT ENOUGH PK:'s AVAILABLE !!!" IF ERR = 6        &
        \GOTO 31000
32 LAST.KB.NO.TRIED% = LAST.KB.NO.TRIED% + 1                      &
        \PKB$ = "PK" + NUM1$(LAST.KB.NO.TRIED%) + ":"             &
        \ON ERROR GOTO 38
36 OPEN PKB$ AS FILE #2%, MODE 1%                                 &
        \GOTO 40
38 RESUME 32 IF ERL = 36 AND ERR = 8                              &
        \PRINT "NOT ENOUGH PK:'s AVAILABLE !!!" IF ERR = 6        &
        \GOTO 31000

!&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
!&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

40 ON ERROR GOTO 50                                               &
        ! Now we have 2 PK:'s opened.  Get this ppn and its       &
        ! password so we can log them into this area.
42 TEMP$=CHR$(6%)+CHR$(14%)+CHR$(0%)+CHR$(SWAP%(0%))+CHR$(0%)+CHR$(0%) &
        \TEMP$=TEMP$+CHR$(0%)+CHR$(0%)+CHR$(1%)+STRING$(13,0)    &
        \TEMP$=TEMP$+STRING$(8,0)                                 &
        \DUMMY$=SYS(TEMP$)
44 A%=ASCII(MID(DUMMY$,9%,1%)) \B%=ASCII(MID(DUMMY$,10%,1%))
45 C%=B%*256%+A% \PASSWORD$=RAD$(C%)
46 ON ERROR GOTO 50                                               &
        \A%=ASCII(MID(DUMMY$,11%,1%)) \B%=ASCII(MID(DUMMY$,12%,1%))
47 C%=B%*256%+A% \PASSWORD$=PASSWORD$+RAD$(C%)
48 ACCOUNT$=NUM1$(ASCII(MID(DUMMY$,8%,1%)))+"/"                   &
        \ACCOUNT$=ACCOUNT$+NUM1$(ASCII(MID(DUMMY$,7%,1%)))        &
        \GOTO 60
50 RESUME 46 IF ERL = 45                                          &
        \RESUME 48 IF ERL = 47                                    &
        \GOTO 31000
```

```
60 ON ERROR GOTO 31000                                          &
        \PRINT #1%, RECORD 1%, "HELLO "; ACCOUNT$; CHR$(13%);   &
        \PRINT #1%, RECORD 1%, PASSWORD$ + CHR$(13%) + CHR$(13); &
        \PRINT #1%, RECORD 1%, "SET VT100; CTRL/R"+CHR$(13)+"X"; &
        \PRINT #2%, RECORD 1%, "HELLO "; ACCOUNT$; CHR$(13%);   &
        \PRINT #2%, RECORD 1%, PASSWORD$ + CHR$(13%) + CHR$(13); &
        \PRINT #2%, RECORD 1%, "SET VT100; CTRL/R"+CHR$(13);    &
        \SLEEP 1%
62 PRESENT.JOB% = 1% \GOSUB 22000
64 PRESENT.JOB% = 2% \GOSUB 22000
68 ON ERROR GOTO 31000                                          &
        \PRINT #1%, RECORD 1%, CHR$(13%);                       &
        \PRESENT.JOB% = 1%                                      &
        \GOTO 100                                               &
        ! The PK:'s should be logged in and ready for use.      &
        ! Now go directly to job #1.


!&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

100 ! Main monitoring loop.....................................
110 GOSUB 20000                                                 &
        \SLEEP 1% IF LEN(CHAR$)=0% AND LEN(DATA.FROM.PK$)=0%    &
        \GOTO 130 IF LEN(CHAR$)=0%
120 IF (CHAR$=CHR$(0%) OR CHAR$=CHR$(28%)) AND LAST.CHAR$=CHR$(0%) &
                GOTO 1000                                       &
        ELSE                                                    &
                LAST.CHAR$=CHAR$
130 GOSUB 2000  ! Put a character to the job and get any data from it.
140 GOTO 100 IF LEN(DATA.FROM.PK$)=0%                           &
        \LSET KB.BUFFER$=DATA.FROM.PK$                          &
        \PUT #10%, RECORD 4096%, COUNT LEN(DATA.FROM.PK$)       &
        \GOTO 100


!&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
!&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

1000 ! Job control routine: control passes here whenever a       &
        ! ^ /^  or  ^ /^\ combination is received.

1010 LSET KB.BUFFER$=CHR$(155%)+"H"+CHR$(155%)+"J"              &
        \PUT #10%, COUNT 4%  ! Clear the screen.

1020 IF ASCII(CHAR$)=28% THEN 1100  ! Try to kill both jobs.

1030 IF PRESENT.JOB% <> 1% THEN                                 &
                PRESENT.JOB% = 1%                               &
        ELSE                                                    &
                PRESENT.JOB% = 2%  ! Switch jobs....
1040 ON ERROR GOTO 1080                                         &
        \PRINT #PRESENT.JOB%, RECORD 6%  ! Check for ^C state.
1050 LAST.CHAR$="" \CHAR$="" \GOSUB 2000                        &
        \GOTO 1070 IF LEN(DATA.FROM.PK$)=0%                     &
        \PRINT #PRESENT.JOB%, RECORD 1%, CHR$(20%);CHR$(18%);   &
        \GOTO 140
1070 PRINT #PRESENT.JOB%, RECORD 1%, CHR$(21%)+"X"+CHR$(13%);   &
        \LAST.CHAR$="" \CHAR$="" \GOTO 120

1080 RESUME 1090
1090 PRINT #PRESENT.JOB%, RECORD 1%, CHR$(20%);                 &
        \LAST.CHAR$=""                                          &
        \CHAR$=CHR$(18%)                                        &
        \GOTO 120  ! Put a ^T ^R combination to the job.

1100 ON ERROR GOTO 1140 \PRINT #1, RECORD 6%  ! Check for ^C state.
1105 FIELD #1%, 128% AS DATA.FROM.PK$                           &
        \LSET DATA.FROM.PK$ = STRING$(128%,0%)
1110 PUT #1%, RECORD 16%, COUNT 0%  ! Kill the job running on channel #1.
1120 GOTO 1200  ! We have killed the first job, now check the 2nd.
1140 RESUME 1150
1150 LSET KB.BUFFER$=CHR$(7%)+CHR$(7%)                          &
        \PUT #10%, COUNT 2%                                     &
        \PRESENT.JOB% = 1%                                      &
        \LAST.CHAR$="" \CHAR$=""                                &
        \GOTO 120

1200 ON ERROR GOTO 1240 \PRINT #2%, RECORD 6%
1205 FIELD #2%, 128% AS DATA.FROM.PK$                           &
        \LSET DATA.FROM.PK$ = STRING$(128%,0%)
1210 PUT #2%, RECORD 16%, COUNT 0%  ! Kill the job on channel #2.
1220 GOTO 30000  ! End the program.....................
1240 RESUME 1250
1250 LSET KB.BUFFER$=CHR$(7%)+CHR$(7%)                          &
        \PUT #10%, COUNT 2%                                     &
        \PRESENT.JOB% = 2%                                      &
        \LAST.CHAR$="" \CHAR$=""                                &
        \GOTO 120

!&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
!&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

2000 ON ERROR GOTO 31000                                        &
        \FIELD #PRESENT.JOB%, 0% AS DATA.FROM.PK$               &
        \GOTO 2010 IF LEN(CHAR$)=0%                             &
        \PRINT #PRESENT.JOB%, RECORD 1%, CHAR$;
2010 ON ERROR GOTO 2050                                         &
        \GET #PRESENT.JOB%, RECORD 8192%, COUNT PK.GET.COUNT%   &
        \FIELD #PRESENT.JOB%, RECOUNT AS DATA.FROM.PK$          &
        \RETURN
2020 RETURN
2050 RESUME 2020 IF ERL = 2010                                  &
        \GOTO 31000


!&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&


20000 ! Routine to get and echo a character from the KB: buffer &
        ! if one is in there.  If not, an empty string is returned.

20020 ON ERROR GOTO 20500 \GET #10%, RECORD 8192%, COUNT 1%     &
        ! Record 8192% says don't wait for input.  Instead, produce &
        !   an error (which we trap in a routine).
20030 !  PUT #10%, COUNT 1%  ! Echo the character as it was received.
20040 CHAR$=LEFT(KB.BUFFER$,1%)  ! Store the character we just got.
20050 RETURN  ! ...from this subroutine.

20500 IF ERL=20020% AND RECOUNT = 0% THEN CHAR$="" \RESUME 20050 &
        ! Here we simply return because we didn't get any data.
20510 GOTO 31000
```

```
!&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

22000   ! Routine to waste output from the present PK:
22010 ON ERROR GOTO 22020 \GET #PRESENT.JOB%, RECORD 8192% \GOTO 22010
22020 SLEEP 1% \RESUME 22030
22030 ON ERROR GOTO 22050 \PRINT #PRESENT.JOB%, RECORD 4% \RETURN
22050 SLEEP 1% \RESUME 22010

!&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

30000 ON ERROR GOTO 30020 \PRINT CHR$(155%);"H";CHR$(155%);"J"
30010 CHAIN "MENU"
30020 GOTO 32767

31000 PRINT "ERROR IN MLTJOB; ERR: ";ERR;"   ERL:";ERL          &
        \RESUME 31010
31010 ON ERROR GOTO 0 \STOP

32767 END

!&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
```



# BASIC IN LONDON

Computer Age Systems, Great Britain, will present "A BASIC Seminar" at the Tara Hotel, Kensington, London, England from March 7, through March 9, 1983.

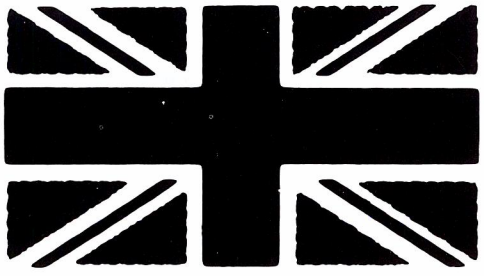
On Monday March 7, Al Cini, president, Computer Methods Corporation, Moorestown, N.J., will present "Structured Programming in DEC BASIC," the first objective analysis of this new product from DEC. Details of RSTS/E Version 8 and the new release of RMS will also be discussed. Cini will include "Structured Programming in BASIC," which he presented at the U.S. DECUS Meetings last year.

On Tuesday, "The Good BASIC Guide to RSTS/E," designed specifically for existing BASIC PLUS—RSTS/E Programmers who want to become more productive, will be offered by Peter Dick, proprietor, Silver Programs, and Chairman of the DECUS UK PDP11 Commercial Users Group. This session will be divided into Simple Functions, Simple Programs, Useful Internals, Never Never Never Use, and The Record Breakers. Documentation will include a listing of all the functions/sub-routines discussed. The code will be available on machine readable format.

Concluding the sessions, on Wednesday some of the U.K.'s leading independent System Suppliers will share their views. Those contributing are Nick Brackenbury, Darkcrest Ltd., "BASIC Licencing"; Al Cini, Computer Methods Corporation, "BASIC Throughput"; Nick de Smith, Computex Ltd., "Inside BASIC"; Sean Seely, ADOS Ltd., "Compiling BASIC"; Michael Stewart, Plan Plus Ltd., "Converting BASIC"; and Peter Wolf, Touche Ross Int., "BASIC Dilemma."

This seminar is not promoted, not authorised, and is not in any way affiliated with Digital Equipment Corporation. Material presented during the seminar in no way reflects specifications or policies of Digital Equipment Corporation.

For information and registration contact Computer Age Systems, P.O. Box 14, Wallingford, Oxon OX10 8NN. ♥



# A RSTS NOSTALGIA

By Charles W. Mustain

A heady bite of ozone, the sharp tang of hot oil, the urgent whine of cooling fans had my heart beating like a kid at Christmas. Four months ago at Parker Street in Maynard I could barely spell 'computer'. Suddenly, I was a system manager.

Adding to the holiday air that November in 1974 were packages of manuals, disk cartridges, blue plastic cases full of gray paper tape and a spool of teletype ribbon I had dropped while trying to install.

I was about to install RSTS-11 Version 4a in my (employer's) shiny new PDP 11/40.

Imagine, if you can, my excitement when I discovered the RSTS-11 manual gave directions for the PDP-11/20 and the PDP-11/45 but seemed never to have heard of the 11/40! After a phone call which assured me there were no important (to RSTS) differences between the 11/40 and the 11/20, I was ready for my next surprise.

The sysgen instructions in the System Manager's Guide detailed methods for building RSTS from DECTAPE, something conspicuously absent from my system. A frantic scan through the manual set (there were only three books) for instructions for a two-RK05 system generation told me I was doomed.

At this point I called frantically for the software specialist I had been assured would do our initial installation. He would be available in about four weeks, when he returned from an on-site consulting job, I was told.

Back to the manuals . . .

A day later and several years older, it had become clear that there were very few differences between DECTAPE and RK05s as far as system generation was concerned. A call to my Parker Street (DEC) instructor helped me figure out how to initialize disks and my service tech gave me the right boot address for the RK05.

For those of you long of tooth and gray of beard, you began building RSTS by starting up a single-user system called DOS. From there you followed directions, learning quickly that if you mistyped anything, you got to start over from the top! Since I was winging it on some of the answers (RK05 instead of DECTAPE) I got almost as much typing practice as a COBOL programmer trying to write a program that prints "HELLO."

There aren't words to tell you the feeling of joy in accomplishment I had when "CAN'T FIND FILE OR ACCOUNT," "READY," clattered out on my Teletype.

Time for my next shock!

I had no DECTAPE #2 with the system library programs. There ensued a frantic attempt to find the library programs on the distribution pack; a search of the box the software came in to see if I had misplaced a disk cartridge; a break for pulling of hair and shedding of tears.

Remember that blue box of gray paper tapes I mentioned? Those were my system library tapes. Someone forgot to tell my salesman that RSTS systems had to include a PC11 high-speed paper tape reader/punch.

On a sheet of paper inserted into one of the manuals was information describing how to read in the library sources through the low-speed (very low speed) reader on the ASR33.

Those of you familiar with ASR33 paper tape readers will remember that the accuracy of these devices was less than perfect. Someone at DEC thoughtfully designed the library tapes and the program that read them in with a scheme whereby checksums were computed for blocks of text on the source tapes.

This, I am sure, was done to make sure the CUSPS (library programs for you greenies to RSTS) would compile and run as intended by Digital. The net effect was that each program had to be run through the ASR33's reader at least three times with crossed fingers and many prayers that it wouldn't drop a bit somewhere, halting the read operation on the last three characters of the tape and forcing one to start over from the beginning of the tape!

Two days later, too exhausted to be thrilled, RSTS-11 V004-A BEREA SCHOOLS IS ON THE AIR, appeared on the TELETYPE, to my intense relief!

♥

# RTS:

# TEST RUNTIME SYSTEM EXAMPLE

By Philip Hunt, O.L.F.B.P., 6400 E. Broad St., Columbus, OH 43213, (614) 864-9200

**Background:**

Runtime systems — Those mythical beings running around your RSTS/E system . . .

During my experiments with RSTS/E, I decided to learn how to write a runtime system since the documentation to do so was at best scarce. Through my trials and tribulations, I discovered the formats for TKB command files, pseudo-vector positions and a whole slew of other items many users might be interested in.

**Description:**

Enclosed is a small Runtime system demonstrating many features and formats required to generate runtime systems. The enclosed runtime system can be added to with very little effort. All that must be done is to add a command to the table called 'CMDTBL' and add the address to jump to to execute the command in dispatch table called 'DISPAT'. When your custom code is complete, just execute a 'JMP RTSINP' to prompt for a new command.

The assemble instructions enclosed also makes the resulting runtime system 'patchable' with ONLPAT. A few examples of patchable areas are included in the source, namely the name printed in response to a 'VERSION' command (at location 'RTSNAM::'), the program name for SYSTAT display (currently 'NONAME' at location PRGNAM::') etc . . .

Currently the runtime system will prompt for input (with 'Ok'), accept a line of input, remove leading spaces, convert it to UPPER CASE. Then if the first character is a '!' or ';', the line is ignored for compatibility with other runtime systems. The command is then checked against the internal commands found in RTS, if not found there, it is executed as a CCL if possible.

Commands currently implemented in RTS follow:

| | | |
|---|---|---|
| RUN | — | Run a program |
| OUT | — | Out to system default KBM |
| VERSION | — | Type version number of RTS |
| ASSIGN | — | Assign devices or logicals |
| DEASSIGN | — | Deassign devices or logicals |
| HELP | — | RTS Help message |

The ASSIGN/DEASSIGN/RUN commands are completely compatible with the format used by Digital-supplied runtime systems.

**Code Description:**

The following is a code-by-code description of RTS.

| SYMBOL STARTING SECTION | DESCRIPTION |
|---|---|
| RTS:: | Runtime system start-up |
| RTSNEW:: | Check if logged out to print 'Bye' if true |
| RTSNME:: | If RTS is entered with a 'SWITCH'-type command, we set the program name to 'NONAME' for SYSTAT. |
| RTSRUN:: | If RTS is entered with a 'RUN' command using a '*.TST' filename, the program name run is setup for SYSTAT. |
| | We then reset the terminal in case a CTRL/C suspended output. |
| | We also set out memory area to 2K. This is not really needed, but included to show example of increasing user memory area. |
| RTSINP:: | We print 'OK'. |
| RTSRED:: | We do some housekeeping such as clearing out our input buffer and core common area. |
| | We wait for input from the user. |
| SPCLOP:: | Drop leading spaces and tabs |
| OKREAD:: | Check for a comment line or just a <CR> or <LF> or any other similar items. |
| CVTLOP:: | Convert lower case to UPPER CASE. |

| | |
|---|---|
| CHECK FOR RTS COMMAND | See if we should execute what was entered as a RTS command |
| CHECK FOR CCL COMMAND | See if we execute as a system CCL command, if so, we do it |
| BADP:: | Tell user it is an illegal input, and prompt for more. |

**Subroutines/Macros Descriptions:**

| | |
|---|---|
| PRSCMD:: | This routine will see if the entered data is a valid RTS command, if so, a flag is set with the index into the dispatch (DISPAT::) table and a flag whether an argument is present. |
| CLRXRB | A macro to clear the user XRB. |
| CLRFQB | A macro to clear the user FIRQB |
| MEMORY | A macro to set the user core size |
| ERROR | A macro to print a RSTS error message on the user terminal |
| MESSAGE | A macro to print a message on the user terminal. |
| TSTFQB | A macro to check any errors occuring after a system call. If there was, the error is printed and a JMP is made to RTSINP for a new prompt and more input. |

**The infamous .99998/.99999 Vectors:**

To generate a runtime system, the user must set up a 'dummy' section for the task-builder which we call '.99998'. A section of code contains pseudo-vector information which we call '.99999'. This section contains the addresses for RSTS to access the RTS for all conditions. More information describing this pseudo-vector region may be found in the RSTS 'System Directives' manual.

The key to runtime system generation is the fact that section '.99998' is expanded with dummy area so that section '.99999' is located at exactly 177732 (octal) for its address. This is what the 'EDIT' mode in the MAKSIL step in the assembly example is for. MAKSIL computes the area required to make .99999 'align' at 177732 and edits your task-build command file to do so.

**Summary:**

The runtime system source enclosed is very straightforward and easily expanded. There really are no limits as to what you can do in a runtime system!!! Just think, a CTRL/C trap that you set up cannot be broken by entering multiple ↑C. As a runtime system, YOU ARE IN CONTROL!!!!

## Questions????

If you are having problems or have questions about RTS or any of my other distributed programs, you may write or call me at the address listed at the top of this article.

A tape containing RTS and all associated files may be yours by sending me $15.00 and a blank tape to the above address.

Happy Computing!!!!!! Until next time!!!!

```
!LOG OF RTS COMPILE/TASK-BUILD/MAKSIL EXECUTION

Ok

!NOTE:RTSDEF.MAC IS MACROS FOR RTS USAGE

MAC RTS,RTS=$COMMON,SY:[1,50]RTSDEF,SY:[1,50]RTS

Ok

!DO THE FIRST TASK-BUILD, NOTE: WE EXPECT MAKSIL TO BELCH AT THIS
!AS IT WILL ENTER EDIT MODE AND FIX UP THE RTS.CMD FILE SO IT IS
!'ALIGNED'


;===================================================================
;******Control file to task-build RTS******
;
RTS/-HD,RTS,RTS=SY:[1,3]RTS
/
;
;THE FOLLOWING 'PAR' STATEMENT WILL ALLOW AD 4K RTS
;NOTE:  IF PHYSICAL MEMORY GOES ABOVE 1K THOUGH, THE STACK PARAMETER WILL
;HAVE TO BE DECREASED (MAKSIL WILL DO THIS AUTOMATICALLY FOR YOU)
;
PAR=RTS:160000:020000
STACK=3072
;
;THE FOLLOWING STATEMENT WILL BE EDITED BY MAKSIL TO EXTEND THE DUMMY
;SECTION TO ALIGN THE RUNTIME SYSTEM, IT CONTAINS NO CODE OR DATA
;
EXTSCT=.99998:0
//
;===================================================================
```



```
TKB @RTS

Ok

!SET UP RTS AS RTS.RTS, NOTE THE /RTS ON THE FIRST COMMAND LINE
!THE EDITED COMMAND FILE WILL BE GENERATED INTO RTS2.CMD

RUN $MAKSIL
MAKSIL  V7.1-11>16K     RSTS V7.1-11 C  OLFBP 11/70
Resident Library name? RTS/RTS
Task-built Run-Time System input file <RTS.TSK>?
The run-time system is not aligned
Edit mode (Yes/No) <Yes>? YES
Task-builder command input file <RTS.CMD>?
The task-builder commands have been changed as follows
  PAR=RTS:160000:020000          PAR=RTS:160000:020000
  STACK=3072                     STACK=3072
  EXTSCT=.99998:0                EXTSCT=.99998:001276

RTS will load in a 4 K-word partition using 1 K-words physical memory.
001276 (octal) bytes may be used for expansion.

Corrected command file name <RTS.CMD>? RTS2
Please task build again using RTS2.CMD

Ok

!RE-TASK-BUILD USING RTS2 WHICH MAKSIL SET UP TO ALIGN THE RTS

TKB @RTS2

Ok

!OK, LETS RUN MAKSIL AGAIN, THIS TIME IT IS ALIGNED SO THE RUNTIME SYSTEM IS
!NOW GENERATED.  NOTE: MAKSIL WILL ALSO DO A 'UT ADD' COMMAND FOR IT.
!NOTE ALSO, THAT WE WANT SYMBOLS (SEE BELOW) SO WE CAN PATCH WITH 'ONLPAT'

RUN $MAKSIL
MAKSIL  V7.1-11>16K     RSTS V7.1-11 C  OLFBP 11/70
Resident Library name? RTS/RTS
Task-built Run-Time System input file <RTS.TSK>?
The run-time system is correctly aligned
Edit mode (Yes/No) <Yes>? NO
Include symbol table (Yes/No) <Yes>?
Symbol table input file <RTS.STB>?
Run-Time System output file <SY:[0,1]RTS.RTS>?
RTS built in 1 K-words, 41 symbols in the directory
RTS.TSK renamed to RTS.TSK<40>

Ok

!ASSEMBLY, TASK-BUILD ARE NOW DONE


Ok

!NOTE BELOW THAT RTS HAS THE FLAGS AUTOMATICALLY SET

SY/R

Run-Time Systems:
 Name   Typ        Size  Users   Comments
BASIC   BAC      16(16)K    2     Perm, Addr:49, KBM, CSZ
CCLMGR  CCL       1(28)K    3     Perm, Addr:192, DF KBM
RSX     TSK       3(28)K    0     Perm, Addr:193, KBM
DCL              12(2)K     0     Non-Res, KBM
BAS2DB  BAC      16(16)K    0     Non-Res, KBM, CSZ
RT11    SAV       4(28)K    2     Temp, Addr:228, KBM, CSZ, EMT:255
RMS11   TSK       4(28)K    0     Non-Res
FOCOMR  DCF      14(16)K    0     Non-Res, Rem
APLSGL  APC      16(16)K    0     Non-Res, KBM
APLDBL  APD      16(16)K    0     Non-Res, KBM
BASIC2  TSK      16(16)K    0     Non-Res
BP2COM  TSK       4(28)K    0     Non-Res, KBM
RTS     TST       1(28)K    0     Non-Res, KBM

Ok

!LETS SWITCH INTO OUT RTS

SW RTS

Ok

!NOTE THAT MY RTS IGNORES EXCLAMATION POINTS
;AND SEMI-COLON LINES IN COLUMN ONE
; THEY ARE CONSIDERED COMMENTS
!THE FOLLOWING ARE RTS COMMANDS, NOT CCLS

HELP
RTS - V01.00.1

Commands:

RUN      - Run a program
OUT      - Out to system default KBM
VERSION  - Type version number of RTS
ASSIGN   - Assign devices or logicals
DEASSIGN- Deassign devices or logicals
HELP     - This message

Ok

VERSION
RTS - V01.00.1
```

. . . continued on page 53

# RSTS PROFESSIONAL CUMULATIVE INDEX

By Greg Justice, Texas Distributors, Inc., Dallas, TX 75234

# DATA BUSINESS LANGUAGE (DBL) IS NOW AVAILABLE IN A VAX NATIVE MODE VERSION — DBL/VAX.

# FOR VAX/VMS

With the addition of DBL/VAX, our DIBOL-11 source code compatible language and compiler is now available for RT-11, TSX/TSX-Plus (time sharing extensions to RT-11), RSTS, RSX-11M/M-Plus, VAX/VMS compatibility mode, and VAX/VMS native mode.

## DBL/VAX FEATURES INCLUDE:

- The DBL/VAX compiler is written in VAX/VMS native mode and is a true compiler.
- Output of the DBL/VAX compiler is in-line native code.
- Multi-user programs can access shared XCALL libraries.
- Entire applications can be "bound" into a single executable module (i.e., an Accounts Payable application).
- Little or NO modification is required to run existing CTS-300 DIBOL code under DBL/VAX native mode.
- DBL/VAX uses the RMS file structure. Those files are then accessible to Datatrieve, FMS, and all other VMS supported languages.
- DBL programs can access and be accessed by other languages.

List price is $5,300.00 and quantity discounts are available to OEM's.
For additional information, please contact:

DISC

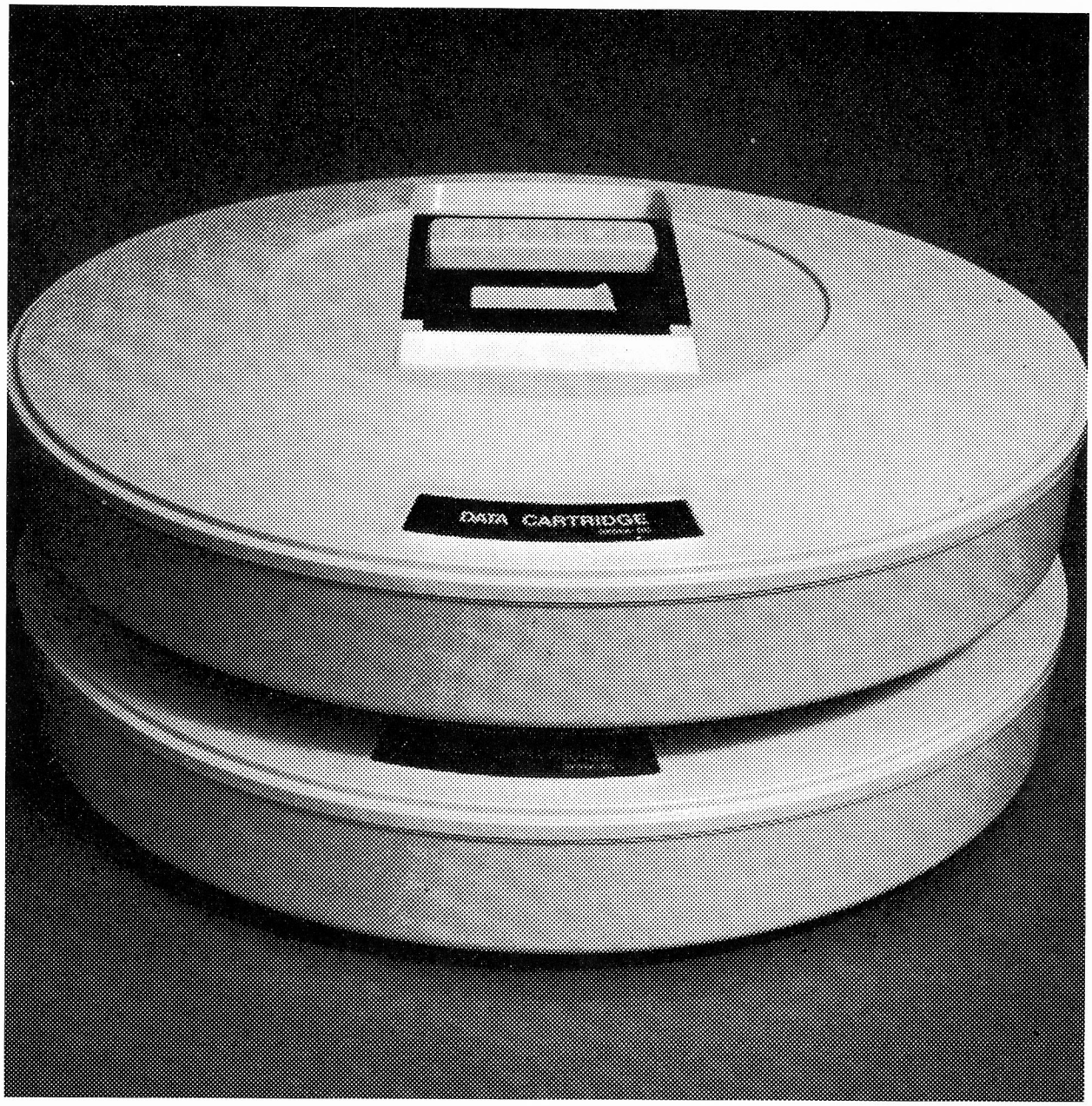
3336 BRADSHAW ROAD SUITE 340 SACRAMENTO, CA 95827 916/363-7385 TWX 910/367-3701
DIGITAL INFORMATION SYSTEMS CORPORATION



CIRCLE 32 ON READER CARD

# The VAX-SCENE

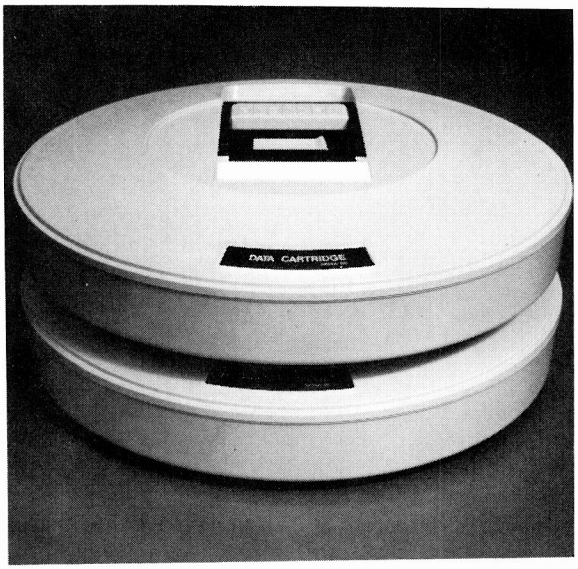
**Number 12** (RSTS PROFESSIONAL, Vol. 5, No. 1) **February 1983**




**INSIDE:**

**SETTING RMS ATTRIBUTES**

**BIG BROTHER** An Automatic Logout Facility for the VAX



# SETTING RMS ATTRIBUTES

By Allen Rueter
Mallinckrodt Institute of Radiology
St. Louis, MO

This is a program that sets RMS attributes to a binary file and then allows you to NFT binary data from RSTS BASIC-PLUS programs to a VAX. You can then read the files without a hassle under VMS.

DEC seems to have cleared up the above problem with Version 3 of VMS.

I've also appended another program which I used to fix the open file and update counts in the UFD back to zero (after a user's program gets zapped by a disk error). The program gives a strange directory listing and prompts for a change when it finds an open file. A return will cause no change. May I suggest that you set up a dummy account to test it, and gain some exposure to how it works.

```
1       extend
5       ! RMS FuDGe                                                     &
        !       This program adds or changes the RMS attributes of a file.  &
        ! By using the /B switch you will set the attributes to FIX=512,    &
        ! which is handy for NFTing binary data files to the VAX.          &

10      print 'This program fudges RMS attributes on to a non RMS file'    &
        \ print ' /B gives you defaults for a binary file'                  &

20      on error go to 32290
30      rf$ = 'UDF FIX VAR VFC STM '            ! Record formats            &
        \ fo$ = 'SEQ REL IDX '                  ! File organization         &

40      dim attribute$(20%)
100     print 'Fudge'; \ input line filename$                               &
        \ filename$ = cvt$$(filename$,38%)      ! remove tabs,spaces,cr,lf,esc &
        \ sw% = instr(1%,filename$,'/') ! any switches ?                    &
        \ switch$ = right(filename$,sw%) if sw%                             &
        \ filename$ = left(filename$,sw%-1%) if sw%                         &

110     if left(switch$,2%) = '/H' then 10000           ! help em           &

120     open filename$ for input as file 1%, mode 1%    ! update            &

130     fs% = num1$(swap%(cvt$%(mid(sys(chr$(12%)),13%,2%))))               &
                                        ! get size of file just opened      &

140     attribute$ = sys(chr$(6%)+chr$(-25%)+chr$(1%)+chr$(0%))             &
                                                ! read file attributes      &
        \ change mid(attribute$,5%,20%) to attribute%   ! get words 1 - 10  &
        \ attribute%(i%) = attribute%(i%*2%-1%)+swap%(attribute%(i%*2%))    &
                        for i%=1% to 10%        ! pack bytes into words     &
        \ if mid(attribute$,5%,20%) <> string$(20%,0%)                      &
                then print filename$;' is all ready a RMS file'             &
                \ rf% = attribute%(1%) and 7%                               &
                \ fo% = (attribute%(1%) and 63%)/8%                         &
                \ rs% = attribute%(2%)                                      &
                \ fs% = attribute%(4%)                                      &
                \ nb% = attribute%(6%)                                      &
                \ nblb% = attribute%(7%)                                    &
                \ bs% = attribute%(8%) and 255%                             &
                \ hs% = swap%(attribute%(8%)) and 255%                      &
                \ mr% = attribute%(9%)                                      &
                \ de% = attribute%(10%)                                     &
                \ print 'RF:';mid(rf$,rf%*4%+1%,3%);                       &
                \ print 'S:';rs%; if rs%                                   &
                \ print ' FO:';mid(fo$,fo%*2%+1%,3%);' USED:';nb%;':';nblb%; &
                                ' RECSI:';mr%;                              &
                \ print                                                     &
                \ input 'Change it any way ';yn$                            &
                \ yn$ = cvt$$(yn$,38%)                                      &
                \ go to 100 if left(yn$,1%) <> 'Y'                          &

160     if left(switch$,2%) = '/B'      ! Make it Binary ?                  &
                then rf% = 1%   ! FIX                                       &
                \ fo% = 0%      ! SEQ                                       &
                \ rs% = 512%                                                &
                \ fs% = val(fs$)                                            &
                \ nb% = fs%                                                 &
                \ nblb% = 512%                                              &
                \ bs%,hs% = 0%                                              &
                \ mr% = rs%                                                 &
                \ go to 400     ! Set it                                    &

200     input 'Record Format:';i$                                           &
        \ i$ = 'FIX' if len(i$) = 0%    ! make default                      &
        \ rec.fmt$ = cvt$$(i$,38%)                                          &
        \ rf% = instr(1%,rf$,rec.fmt$)                                      &
        \ go to 200 if rf% = 0%                                             &
        \ rf% = (rf%-1%)/4%                                                 &

220     input 'File organization:';i$                                       &
        \ i$ = 'SEQ' if len(i$) = 0%    ! make default                      &
        \ fil.org$ = cvt$$(i$,38%)                                          &
        \ fo% = instr(1%,fo$,fil.org$)                                      &
        \ go to 220 if fo% = 0%                                             &
        \ fo% = ( (fo%-1%)/2%) * 8%                                         &

240     input 'Recordsize:';i$                                              &
        \ i$ = '512' if len(i$) = 0%                                        &
        \ rs% = val(i$)                                                     &

260     input 'File size:';fs%                                              &

280     input 'No of blocks in use:';nb%
300     input 'No of bytes in last block:';i%                               &
        \ i$ = '512' if len(i$) = 0                                         &
        \ nblb% = val(i$)                                                   &

320     input 'Bucket size:';bs%                                            &

340     input 'Header size:';hs%                                            &

360     input 'Maximum record size:';i$                                     &
        \ i$ = '512' if len(i$) = 0                                         &
        \ mr% = val(i$)                                                     &

380     input 'Default Extend size in blocks:';de%
400     s$ = sys(chr$(6%)+chr$(-25%)+chr$(1%)+chr$(10%)+        ! write 10 words &
                cvt$$(swap%(rf%+fo%))+          ! word 1 - rec fmt & fil org  &
                cvt$$(swap%(rs%))+              ! word 2 - rec size           &
                cvt$$(swap%(0%))+cvt$$(swap%(fs%))+     ! file size (in blocks) &
                cvt$$(swap%(0%))+cvt$$(swap%(nb%))+     ! number of blocks in use &
                cvt$$(swap%(nblb%))+    ! number of bytes in last block       &
                chr$(bs%)+chr$(hs%)+    ! bucket size, header size            &
                cvt$$(swap%(mr%))+      ! max rec size                        &
                cvt$$(swap%(de%)) )     ! default extention size              &

420     go to 100                                                           &

10000   print 'Enter a file name for which you would like to add or change RMS attributes.' &
        \ print                                                             &
        \ print '       Switches are:'                                      &
        \ print '               /B - Set Binary, Seq. fixed 512 byte records' &
        \ print '                       Need to NFT data files to a Files-11 system(RSX,VMS).' &
        \ print                                                             &
        \ print '       Control Z to Exit.'                                 &
        \ print                                                             &
        \ go to 100                                                         &

32290   if err = 11 and erl = 100 then 32767
32300   if erl = 120 then print right(sys(chr$(6%)+chr$(9%)+chr$(err)),3%)  &
                \ resume 100                                                &

32700   on error go to 0
32767   end
```

```
1       extend
10      ! This program is mainly for clearing the open file count and open  &
        ! update mode bit after your program gets a swap error or other such &
        ! annoying problems.                                               &
        !       For other goodies see Scott Banks article in Sep 80 RSTS Pro &
        !       page 38                                                   &
        !                                                                 &
        !       Name Blockette                                            &
        ! 0     Link to next name blockette                               &
        ! 1     Filename rad50  (-1 for UFD in MFD)                        &
        ! 2     Filename rad50                                            &
        ! 3     Extension rad50                                           &
        ! 4     Protection code / status                                  &
        ! 5     Read/Only open count / Open file count                    &
        ! 6     Link to accounting Blockette (project # / programmer # in UFD) &
        ! 7     Link to 1st retrieval Blockette (-31692 = 'UFD'Rad50)     &

20      ! S T A T U S  B Y T E                                             &
        ! 7  6  5  4  3  2  1  0                                           &
        ! ------------------------                                        &
        !128 64 32 16 8  4  2  1                                          &
        !------------------------                                        &
        ! !  !  !  !  !  !  !  !                                          &
        ! !  !  !  !  !  !  !  !-- File is out of sat                     &
        ! !  !  !  !  !  !  !----- File is placed                         &
        ! !  !  !  !  !  !------- Write acces given out                  &
        ! !  !  !  !  !---------- File open in UPDATE mode                &
        ! !  !  !  !------------- No file extending allowed               &
        ! !  !  !---------------- No delete and/or rename allowed          &
        ! !  !------------------- Entry is MFD entry                       &
        ! !---------------------- File marked for deletion                 &

50      print 'Enter UFD to be cleaned as [*,*]';                          &
        \ input line ufd$                                                 &
        \ ufd$ = cvt$$(ufd$,4%)                                           &

100 open ufd$ for input as file 1%, mode 16384%
105 dim #1%, u%(3583%,7%)
120     clu% = u%(31%,0%)
200     go sub 1000             ! list directory                          &

300     go to 32767
1000    nb%,ptr% = fnlink%(u%(0%,0%))   ! find 1st name blockett          &

1020    go to 1190 unless ptr%          ! if null then exit               &
        \ go sub 2000                   ! print this nb                   &
        \ ptr% = fnlink%(u%(ptr%,0%))   ! link to next                    &
        \ go to 1020                                                      &

1190    return                                                            &

2000    print rad$(u%(ptr%,1%))+rad$(u%(ptr%,2%));      ! prnt file nam   &
                '.';rad$(u%(ptr%,3%));                  ! & extension     &

2020    print using' <###>', swap%(u%(ptr%,4%)) and 255%;        ! prot   &

2040    ab% = fnlink%(u%(ptr%,6%))                                        &
\       print using ' ##### ###', u%(ab%,2%);   ! file size in blocks     &
                u%(ab%,7%);                     ! file clustersize        &

2050    print using ' ### ###', u%(ptr%,4%) and 255%; u%(ptr%,5%);        &

2060    print                                                             &

2070    goto 2090 if u%(ptr%,5%) = 0%                                     &
\       yn$ = ''                                                          &
\       Input 'Clear the Update bit(y/n)';yn$  if u%(ptr%,4%) and 8%      &
\       yn$ = cvt$$(yn$,32%)                                              &
\       u%(ptr%,4%)=u%(ptr%,4%) and -9% if left(yn$,1%) = 'Y'             &
\       yn$ = ''                                                          &
\       Input 'Clear the write access bit';yn$ if u%(ptr%,4%) and 4%      &
\       yn$ = cvt$$(yn$,32%)                                              &
\       u%(ptr%,4%)=u%(ptr%,4%) and -5% if left(yn$,1%) = 'Y'             &
\       yn$ = ''                                                          &
\       Input 'Clear the mark for deletion bit';yn$ if u%(ptr%,4%) and 128% &
\       yn$ = cvt$$(yn$,32%)                                              &
\       u%(ptr%,4%) = u%(ptr%,4%) and -129% if left(yn$,1%) = 'Y'         &

2080    print 'read only count =';swap%(u%(ptr%,5%)) and 255%            &
                'Open count =';u%(ptr%,5%) and 255%                       &
\       nrc% = swap%(u%(ptr%,5%)) and 255%                                &
\       oc% = u%(ptr%,5%) and 255%                                        &
\       Input 'New read only count';nrc$ \ nrc% = val(nrc$) if len(nrc$)  &
\       Input 'New open count';noc$ \ oc% = val(noc$) if len(noc$)        &
\       u%(ptr%,5%) = swap%(nrc% and 255%) or (oc% and 255%)              &

2090    Return                                                            &

10000   def fnlink%( 1% )               ! cvt raw ptr to virt array ptr   &

10010   ! bits 15-12 Block within cluster (*4096%)                        &
        ! bits 11- 9 Cluster              (*512%)                         &
        ! bits  8- 4 blockette within block (*16%)                        &
        ! bits  3- 0 special bit flags     (*1)                           &

10020   cluster% = ( 1% and 3584% ) / 512%                                 &
        \ blck% = swap%( 1% and -4096% ) / 16%                            &
        \ blockette% = ( 1% and 496% ) / 16%                              &
        \ fnlink% = (cluster% * clu% + blck%) * 32% + blockette%
10030   fnend                                                             &

32767   end
```

# BIG BROTHER

## An Automatic Logout Facility for the VAX

By Niall McPhillips, Petroconsultants Ltd., Ireland

An unattended terminal left logged in poses a security risk to any computer system. Many systems have an automatic logout feature which logs out a user whose terminal has been idle for a period of time. Unfortunately this is a feature which VMS doesn't and, according to DEC's software dispatches, won't have.

BIG-BROTHER is just such a program; it will log out users who have not used any system resources for a given time. It will not, however, stop any process which is running an executable image, even if that process has been idle, as this could cause problems with any open files. Written in VAX PL/1 it runs in this installation under VMS V3. If you haven't got a PL/1 compiler don't despair, as it would be relatively simple to write a similar program based on the principles outlined below in any other language supported by VMS.

The program scans through all the processes on the system at regular intervals and requests the following information for each process:—

1) Process ID.
2) CPU time to data.
3) Name and name length of the image currently running.
4) Group no. of process UIC.
5) Subprocess count.
6) Terminal identifier.

If no image is running (image name length of 0), if the group number of the UIC is greater than one (i.e., not a system process), and no subprocesses are currently active then the process ID, CPU time and terminal are stored in a list of idle processes. This is then compared against the last list taken. Any process which appears on both lists with an unchanged CPU time is deleted and an appropriate message is output to the terminal. A wakeup is then scheduled to occur after time DELTA-TIME and the program hibernates until then. In this installation we use 10 minutes as the delta-time, but this can be easily changed if required.

You may want to customize the program for your particular installation. For instance, you may wish to exclude certain terminals or users from being logged out, or you may wish to hold a log file of all processes logged out (to discover the culprits who most often leave their terminals unattended). These can be easily added to the program by, in the first case, adding conditions excluding your desired UICs/terminals to the conditions to be satisfied before the process is put on to the 'idle list'; and in the second case, all that is required is for a record containing the process information to be output to a log-file as the process is deleted.

BIG BROTHER is best run as a detached process which is activated at system startup and left running permanently. Since it only uses resources briefly every 10 minutes it has little or no effect on system performance.

```
BIG_BROTHER: PROCEDURE OPTIONS (MAIN) ;
/*
    This is a program to automatically log off terminals which
    have been idle for a time.

    To do this it compiles information at ten minute intervals
    on all processes running.

    If a process
              i)  Is not running a program (Image name length of 0).
          & ii)  Has been idle (CPU time not changed since last inspection).
          & iii) Has no subprocesses running (Subprocess count of 0).
          & iv)  Has a group no. greater than 1 (Is not a system process).

    then it will be stopped and an appropriate message will be
    output to whatever terminal it was using.
*/

        %INCLUDE SYS$GETJPI ;
        %INCLUDE SYS$DELPRC ;
        %INCLUDE SYS$BINTIM ;
        %INCLUDE SYS$SCHDWK ;
        %INCLUDE SYS$HIBER ;

        %REPLACE NO_PROCESSES BY 50 ;
        %REPLACE TRUE  BY '1'B ;
        %REPLACE FALSE BY '0'B ;

        DECLARE 1 JPI_LIST STATIC EXTERNAL,   /* List structure for SYS$GETJPI */
                  2 JPI_CPUTIM,       /* CPU time */
                    3 LENGTH  FIXED BINARY (15) INIT (4),
                    3 CODE    FIXED BINARY (15) INIT (JPI$_CPUTIM),
                    3 ADDRESS POINTER,
                    3 RET_LEN FIXED BINARY (31) INIT (0),
                  2 JPI_IMAGE,        /* Image name */
                    3 LENGTH  FIXED BINARY (15) INIT (128),
                    3 CODE    FIXED BINARY (15) INIT (JPI$_IMAGNAME),
                    3 ADDRESS POINTER,
                    3 RET_LEN POINTER,
                  2 JPI_GROUP,        /* Group no. */
                    3 LENGTH  FIXED BINARY (15) INIT (4),
                    3 CODE    FIXED BINARY (15) INIT (JPI$_GRP),
                    3 ADDRESS POINTER,
                    3 RET_LEN FIXED BINARY (31) INIT (0),
                  2 JPI_PROCID,       /* Process ID */
                    3 LENGTH  FIXED BINARY (15) INIT (4),
                    3 CODE    FIXED BINARY (15) INIT (JPI$_PID),
                    3 ADDRESS POINTER,
                    3 RET_LEN FIXED BINARY (31) INIT (0),
                  2 JPI_TERM,         /* Terminal identifier */
                    3 LENGTH  FIXED BINARY (15) INIT (7),
                    3 CODE    FIXED BINARY (15) INIT (JPI$_TERMINAL),
                    3 ADDRESS POINTER,
                    3 RET_LEN FIXED BINARY (31) INIT (0),
                  2 JPI_SUBPRC,       /* Subprocess count */
                    3 LENGTH  FIXED BINARY (15) INIT (4),
                    3 CODE    FIXED BINARY (15) INIT (JPI$_JOBPRCCNT),
                    3 ADDRESS POINTER,
                    3 RET_LEN FIXED BINARY (31) INIT (0),
                  2 ENDLIST   FIXED BINARY (31) INIT (0) ;

       DECLARE (SUBPROC, GROUP_NO)  FIXED BINARY (31),
               (PID, NAMLEN, ISTAT) FIXED BINARY (31),
               (I, J, INDEX, CPUTIM)FIXED BINARY (31),
               ID                   FIXED BINARY (31),
               BINARY_DELTA_TIME    BIT (64) ALIGNED,
               DELTA_TIME           CHAR (13) INIT ('0 00:10:00.00'),
               TERM_READY           BIT,
               IMAGE_NAME           CHARACTER (128),
               OUT_TERM             FILE PRINT,
               PROC_TERM            CHARACTER (7),
               TERMINALS (50)       CHARACTER (7) INIT ((50)('       ')) ;

       DECLARE (SS$_NORMAL, SS$_NOMOREPROC) FIXED BINARY (31) GLOBALREF VALUE ;

       DECLARE OUT_MSG CHAR (50) INIT ('   User logged off  - this terminal is now free  !!') ;

       DECLARE (LAST_PROCESSES (50), CURR_PROCESSES (50), LAST_CPUTIM (50),
                CURR_CPUTIM (50))    FIXED BINARY (31) INIT ((50)-1) ;

    /* Set up the addresses for the list structure */

       JPI_CPUTIM.ADDRESS = ADDR (CPUTIM) ;
```

```
        JPI_IMAGE.ADDRESS  = ADDR (IMAGE_NAME) ;
        JPI_IMAGE.RET_LEN  = ADDR (NAMLEN) ;
        JPI_GROUP.ADDRESS  = ADDR (GROUP_NO) ;
        JPI_PROCID.ADDRESS = ADDR (PID) ;
        JPI_TERM.ADDRESS   = ADDR (PROC_TERM) ;
        JPI_SUBPRC.ADDRESS = ADDR (SUBPROC) ;

    /* Start the infinite loop */

        DO WHILE ( TRUE ) ;

           INDEX = 1 ;  /* Counter for arrays */
           ISTAT = SS$_NORMAL ;
           ID = -1 ;

       /* Go through all the processes that we can get info on */

           DO WHILE ( (ISTAT ^= SS$_NOMOREPROC) & (INDEX <= NO_PROCESSES) ) ;

              ISTAT = SYS$GETJPI (,ID,,JPI_LIST,,,) ;  /* Get the info on the next process */

              IF ISTAT = SS$_NORMAL    /* All is OK ? */
                 THEN DO ;
                       IF (NAMLEN = 0) & (GROUP_NO > 1) & (SUBPROC = 0)
                          THEN DO ;   /* Set up the arrays */
                                CURR_PROCESSES (INDEX) = PID ;
                                CURR_CPUTIM (INDEX) = CPUTIM ;
                                TERMINALS (INDEX) = PROC_TERM ;
                                INDEX = INDEX + 1 ;
                               END ;
                    END ;

           END ;  /* No more processes - all have been examined */

    /*   We now know all processes which are currently
         doing nothing - now see if they were doing
         nothing the last time we looked.                  */

           DO I = 1 TO INDEX ;

              J = 1 ;
              DO WHILE ( LAST_PROCESSES(J) ^= -1 ) ;
                 IF CURR_PROCESSES(I) = LAST_PROCESSES(J) THEN
                   IF CURR_CPUTIM(I) = LAST_CPUTIM(J) THEN DO ;

                       /* Delete the process & output a message */

                      ISTAT = SYS$DELPRC (CURR_PROCESSES(I),) ;

                      ON UNDEFINEDFILE (OUT_TERM) TERM_READY = FALSE ;

                      TERM_READY = FALSE ;  /* Loop until terminal is available */
                      DO WHILE ( ^TERM_READY ) ;
                         TERM_READY = TRUE ;
                         OPEN FILE (OUT_TERM) OUTPUT TITLE (TERMINALS(I)) ;
                      END ;

                      PUT FILE (OUT_TERM) LIST (OUT_MSG) ;
                      CLOSE FILE (OUT_TERM) ;

                    END ;
                 J = J + 1 ;
              END ;  /* End of DO WHILE */

           END ;  /* End of outer do loop ( 1 to INDEX ) */

       /* Now set up the arrays for the next loop */

           DO I = 1 TO NO_PROCESSES ;
              LAST_PROCESSES (I) = CURR_PROCESSES (I) ;
              LAST_CPUTIM (I) = CURR_CPUTIM (I) ;
              CURR_PROCESSES (I) = -1 ;
              CURR_CPUTIM (I) = -1 ;
              TERMINALS (I) = '        ' ;
           END ;

       /* Hibernate the process for a time */

           ISTAT = SYS$BINTIM (DELTA_TIME, BINARY_DELTA_TIME) ;
           ISTAT = SYS$SCHDWK (,,BINARY_DELTA_TIME,) ;
           ISTAT = SYS$HIBER () ;

        END ;   /* End of the infinite DO WHILE loop */

END BIG_BROTHER ;
```

# RTS: Test Runtime System Example

. . . continued from page 42

```
Ok

OUT

Ok


!NOTE: ABOVE CCLMGR IS AN EXPANDED VERSION OF RTS ALLOWING INTERFACE TO
!THE CCLMAN FILE PUBLISHED IN THE JUNE, 1982 ISSUE OF RSTS PROFESSIONAL.
!THIS ALLOWS ALL CCLMAN CCLS TO BE EXECUTED AS IF THEY WERE REAL CCLS. NOT
!REQUIRING '@@ ' TO BE APPENDED.
!
!


!LOG OF RTS COMPILE/TASK-BUILD/MAKSIL EXECUTION

Ok

!NOTE:RTSDEF.MAC IS MACROS FOR RTS USAGE

MAC RTS,RTS=$COMMON,SY:[1,50]RTSDEF,SY:[1,50]RTS

Ok

!DO THE FIRST TASK-BUILD, NOTE: WE EXPECT MAKSIL TO BELCH AT THIS
!AS IT WILL ENTER EDIT MODE AND FIX UP THE RTS.CMD FILE SO IT IS
!'ALIGNED'


;====================================================================
;******Control file to task-build RTS******
;
RTS/-HD,RTS,RTS=SY:[1,3]RTS
/
;
;THE FOLLOWING 'PAR' STATEMENT WILL ALLOW AD 4K RTS
;NOTE:  IF PHYSICAL MEMORY GOES ABOVE 1K THOUGH, THE STACK PARAMETER WILL
;HAVE TO BE DECREASED (MAKSIL WILL DO THIS AUTOMATICALLY FOR YOU)
;
PAR=RTS:160000:020000
STACK=3072
;
;THE FOLLOWING STATEMENT WILL BE EDITED BY MAKSIL TO EXTEND THE DUMMY
;SECTION TO ALIGN THE RUNTIME SYSTEM, IT CONTAINS NO CODE OR DATA
;
EXTSCT=.99998:0
//
;====================================================================


TKB @RTS

Ok

!SET UP RTS AS RTS.RTS, NOTE THE /RTS ON THE FIRST COMMAND LINE
!THE EDITED COMMAND FILE WILL BE GENERATED INTO RTS2.CMD

RUN $MAKSIL
MAKSIL  V7.1-11>16K     RSTS V7.1-11 C  OLFBP 11/70
Resident Library name? RTS/RTS
Task-built Run-Time System input file <RTS.TSK>?
The run-time system is not aligned
Edit mode (Yes/No) <Yes>? YES
Task-builder command input file <RTS.CMD>?
The task-builder commands have been changed as follows
  PAR=RTS:160000:020000           PAR=RTS:160000:020000
  STACK=3072                      STACK=3072
  EXTSCT=.99998:0                 EXTSCT=.99998:001276

RTS will load in a 4 K-word partition using 1 K-words physical memory.
001276 (octal) bytes may be used for expansion.

Corrected command file name <RTS.CMD>? RTS2
Please task build again using RTS2.CMD

Ok

!RE-TASK-BUILD USING RTS2 WHICH MAKSIL SET UP TO ALIGN THE RTS

TKB @RTS2

Ok

!OK, LETS RUN MAKSIL AGAIN, THIS TIME IT IS ALIGNED SO THE RUNTIME SYSTEM IS
!NOW GENERATED.  NOTE: MAKSIL WILL ALSO DO A 'UT ADD' COMMAND FOR IT.
!NOTE ALSO, THAT WE WANT SYMBOLS (SEE BELOW) SO WE CAN PATCH WITH 'ONLPAT'

RUN $MAKSIL
MAKSIL  V7.1-11>16K     RSTS V7.1-11 C  OLFBP 11/70
Resident Library name? RTS/RTS
Task-built Run-Time System input file <RTS.TSK>?
The run-time system is correctly aligned
Edit mode (Yes/No) <Yes>? NO
Include symbol table (Yes/No) <Yes>?
Symbol table input file <RTS.STB>?
Run-Time System output file <SY:[0,1]RTS.RTS>?
RTS built in 1 K-words, 41 symbols in the directory
RTS.TSK renamed to RTS.TSK<40>

Ok
```

```
!ASSEMBLY, TASK-BUILD ARE NOW DONE

Ok

!NOTE BELOW THAT RTS HAS THE FLAGS AUTOMATICALLY SET

SY/R

Run-Time Systems:
 Name   Typ       Size  Users   Comments
BASIC   BAC     16(16)K   2     Perm, Addr:49, KBM, CSZ
CCLMGR  CCL      1(28)K   3     Perm, Addr:192, DF KBM
RSX     TSK      3(28)K   0     Perm, Addr:193, KBM
DCL              12(2)K   0     Non-Res, KBM
BAS2DB  BAC     16(16)K   0     Non-Res, KBM, CSZ
RT11    SAV      4(28)K   2     Temp, Addr:228, KBM, CSZ, EMT:255
RMS11   TSK      4(28)K   0     Non-Res
FOCOMR  DCF     14(16)K   0     Non-Res, Rem
APLSGL  APC     16(16)K   0     Non-Res, KBM
APLDBL  APD     16(16)K   0     Non-Res, KBM
BASIC2  TSK     16(16)K   0     Non-Res
BP2COM  TSK      4(28)K   0     Non-Res, KBM
RTS     TST      1(28)K   0     Non-Res, KBM

Ok

!LETS SWITCH INTO OUT RTS

SW RTS

Ok

!NOTE THAT MY RTS IGNORES EXCLAMATION POINTS
;AND SEMI-COLON LINES IN COLUMN ONE
; THEY ARE CONSIDERED COMMENTS
!THE FOLLOWING ARE RTS COMMANDS, NOT CCLS

HELP
RTS - V01.00.1

Commands:

RUN     - Run a program
OUT     - Out to system default KBM
VERSION - Type version number of RTS
ASSIGN  - Assign devices or logicals
DEASSIGN- Deassign devices or logicals
HELP    - This message

Ok

VERSION
RTS - V01.00.1

Ok

OUT

Ok

!NOTE: ABOVE CCLMGR IS AN EXPANDED VERSION OF RTS ALLOWING INTERFACE TO
!THE CCLMAN FILE PUBLISHED IN THE JUNE, 1982 ISSUE OF RSTS PROFESSIONAL.
!THIS ALLOWS ALL CCLMAN CCLS TO BE EXECUTED AS IF THEY WERE REAL CCLS, NOT
!REQUIRING '@@ ' TO BE APPENDED.
!
!
```

```
        .TITLE  RTSDEF,RTS Definition Macros,
.NLIST MD,ME,MEB

; RTS SYSTEM MACRO DEFINITION FILE
; (C) 1982, OLFBP       Philip Hunt
;

.MACRO  CLRFQB
        CALL    $FRBCL
        .ENDM

.MACRO  CLRXRB
        CALL    $XRBCL
        .ENDM

.MACRO  ERROR   NUM
        PUSH    R1
        MOVB    NUM,R1
        CALL    $ERRPT
        POP     R1
        .ENDM

.MACRO  MESSAGE MSG,LEN
        CLRXRB
        MOV     LEN,XRB+XRLEN
        MOV     XRB+XRLEN,XRB+XRBC
        MOV     MSG,XRB+XRLOC
        .WRITE                          ;WRITE MESSAGE PROMPT
        .ENDM

.MACRO  INPUT   BUF,ARG
        MOV     ARG,XRB+XRLEN
        MOV     BUF,XRB+XRLOC
        MOV     #-1,XRB+XRTIME          ;^C STATE ON ^T
        .READ                           ;GET USER INPUT
        .ENDM

.MACRO  MEMORY  ARG
        CLRXRB
        MOVB    ARG,XRB+0
        .CORE
        .ENDM

.MACRO  TSTFQB
        TSTB    FIRQB
        BEQ     10$
        CALL    $PRFRQ
        JMP     RTSINP
10$:
```

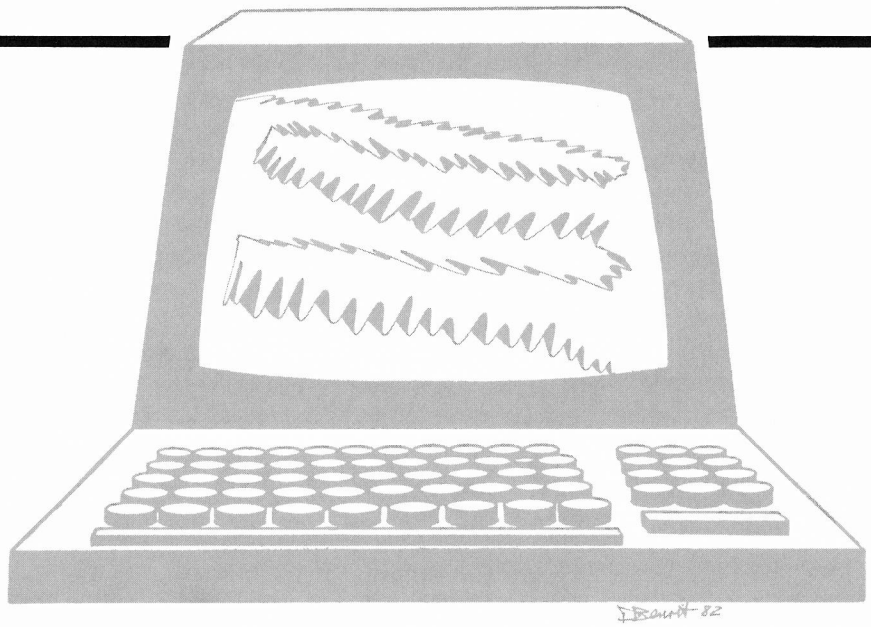
---

```
 1                              .TITLE  RTSDEF,RTS Definition Macros,01,23-Jun-82,PJH
 2                      .NLIST MD,ME,MEB
 3
 4                      ; RTS SYSTEM MACRO DEFINITION FILE
 5                      ; (C) 1982, OLFBP       Philip Hunt
 6                      ;
 7
11
15
22
30
37
43
```

```
 1                              .TITLE  RTS,RTS Test Runtime System,01,23-Jun-82,PJH
 2
 3                      ;       *****RTS***** TEST RUNTIME SYSTEM EXAMPLE
 4                      ;
 5                      ;       (C) 1982        OLFBP           Philip Hunt
```

```
 6
 7 000000                               .PSECT  RTSMON,RO,GBL
 8                                       .ENABL  GBL
 9                                       .NLIST  ME
10                                     ;
11                                     ;LOW CORE DATA AREA DEFINITIONS
12                                     ;
13         001002                      BUFFER = 1002                          ;PUT BUFFER BOTTOM USERSP
14
15                                     ;       COPY    $COMMON
16                                     ;       COPY    RTSDEF                 ;MACRO DEF INCLUDES
17
18                                     ; start-up RTS, initialize items, set system program name
19
20 000000                              RTS::
21 000000                                       CLRXRB                         ;HOUSEKEEPING
22 000004                                       CLRFQB
23 000010  000423                               BR      RTSNME                 ;NO LOG OUT CHECK
24
25 000012  016702  000400           RTSNEW: MOV     KEY,R2                 ;CHECK KEYWORD JOB BITS
26 000016  032702  010000                   BIT     #JFNOPR,R2             ;SEE IF LOGGED OUT
27 000022  001416                           BEQ     RTSNME                 ;LOGGED IN
28
29 000024                                       MESSAGE #BYEMSG,#8.            ;print 'bye'
30 000054  000167  000070                       JMP     RTSRED                 ;GET A COMMAND
31
32 000060  016767  001672  000412   RTSNME: MOV     PRGNAM,FIRQB+FQNAM1
33 000066  016767  001666  000414           MOV     PRGNAM+2,FIRQB+FQNAM1+2
34
35 000074  104044                    RTSRUN::.NAME                          ;NAME US AS "NONAME"
36 000076  104026                            .TTRST                         ;JUST IN CASE, ENTRY BY ^C
37
38 000100                                       MEMORY  #2                     ;GET 2K WORDS
39
40                                     ; print prompt, initialize buffers
41
42 000114                              RTSINP::CLRXRB
43 000120                                       MESSAGE #PROMPT,#7.            ;PRINT PROMPT
44
45 000150  012701  000100           RTSRED: MOV     #64.,R1
46 000154  012702  000460                   MOV     #CORCMN,R2             ;LETS CLEAR CORCOMMON AREA
47 000160  012703  001002                   MOV     #BUFFER,R3
48 000164  005022                   CORLOP: CLR     (R2)+
49 000166  005023                           CLR     (R3)+                  ;ALSO THE INPUT BUFFER
50 000170  077103                           SOB     R1,CORLOP
51
52                                     ; get user input
53
54 000172                                       CLRXRB
55 000176                                       INPUT   #BUFFER,#128.          ;GET USER INPUT
56 000222                                       TSTFQB                         ;ANY ERRORS???
57
58                                     ; do asthetic housekeeping....
59
60 000240  016705  000444                   MOV     XRB+XRBC,R5            ;SAVE CHARS ACTUALLY RECIEVED
61 000244  012704  001002                   MOV     #BUFFER,R4             ;GET BUFFER RCVD ADDR
62 000250  121427  000040           SPCLOP: CMPB    @R4,#32.               ;SPACE AS FIRST CHARACTER?
63 000254  001403                           BEQ     DROPCH                 ;YES, SO DROP IT
64 000256  121427  000011                   CMPB    @R4,#9.                ;TAB AS FIRST CHAR???
65 000262  001002                           BNE     OKREAD                 ;NOPE, MUST BE OK...
66
67 000264  005204                   DROPCH::INC     R4                     ;YES, SO SKIP IT
68 000266  000770                           BR      SPCLOP                 ;AND TRY AGAIN
69
70 000270  122714  000015           OKREAD::CMPB    #13.,@R4               ;CR ONLY?????
71 000274  001725                           BEQ     RTSRED                 ;YEP, SO IGNORE IT
72 000276  122714  000012                   CMPB    #10.,@R4               ;LF ONLY?????
73 000302  001722                           BEQ     RTSRED                 ;YEP, SO IGNORE IT
74 000304  122714  000004                   CMPB    #04.,@R4               ;ctrl/d ONLY?????
75 000310  001717                           BEQ     RTSRED                 ;YEP, SO IGNORE IT
76 000312  122714  000014                   CMPB    #12.,@R4               ;ctrl/l ONLY?????
77 000316  001714                           BEQ     RTSRED                 ;YEP, SO IGNORE IT
78 000320  122714  000033                   CMPB    #27.,@R4               ;ESC ONLY?????
79 000324  001711                           BEQ     RTSRED                 ;YEP, SO IGNORE IT
80 000326  122714  000041                   CMPB    #33.,@R4               ;EXCLAMATION, SO ASSUME A COMMENT
81 000332  001706                           BEQ     RTSRED                 ;YES, IGNORE IT
82 000334  122714  000073                   CMPB    #59.,@R4               ;SEMI-COLON,  SO ASSUME A COMMENT
83 000340  001703                           BEQ     RTSRED                 ;YES, IGNORE IT
84
85                                     ; lower to upper case conversion
```

. . . continued on page 60



# LINE NUMBER RESEQUENCER FOR BASIC-PLUS AND B+II PROGRAMS

By Lawrence P. Gallagher

Resequencers are programs which renumber the lines of a BASIC source file. This function facilitates the addition of new sub-routines and the linkage of several sub-programs to a main source. Also, resequencers modify the line number arguments of GOTO's and other similar statements, to conform to the new line sequence.

There are several undesirable features in the DEC supplied RESEQ.BAC (VER 3B-01). First, it does not process programs with ampersand-flagged multi-line commands; these files it hashes beyond recognition. Furthermore, RESEQ.BAC does not back up the file it is processing, making error recovery virtually impossible. Lastly, there is a maximum program length allowed by RESEQ.BAC, which is inconvenient when trying to concatenate several large programs.

RESEQ.TEC (V01), however, has none of these deficiencies. The TECO run-time system has a unique file opening mode ("/B+" mode) which recognizes ampersand-flagged statements in a BASIC source file. TECO also has an inherent "OPEN and BACKUP" command. TECO employs a variable length text buffer and internal stack along with a variety of commands such as INSERT, SEARCH, and SUBSTITUTE, and TECO can handle exceptionally large files by splitting them into pages. These features make TECO an ideal language for resequencers.

When RESEQ.TEC is run, two macros are defined and loaded into their respective Q-registers: a terminal driver into QB, and a "line number lookup and substitute" macro into QR. RESEQ.TEC then prompts the user to enter his file name (which defaults to a ".BAS" extension) until his file can be found. After the file is opened (in "/B+" mode), the user is prompted to enter the line number parameters: the lowest and highest line numbers of the original program segment, and the starting number and interval size of the new program lines.

During the first pass of resequencing, RESEQ.TEC successively scans each line of the source file looking for those lines whose line numbers are within the range specified by the user. If the number is in range, RESEQ.TEC loads the old line number in the numeric storage space of QT, computes the corresponding new line number, and loads the new line number in the text storage space of QT. QT is then pushed on the stack, and the new line number counter is incremented. (If by some chance the newly computed line numbers overflow, or become greater than 32767, RESEQ.TEC prints a warning, and aborts, restoring the original program.) After the entire program has been scanned, the entire stack is popped into the now-empty text buffer in table form, and the entire table is stored in the text storage area of QX.

RESEQ.TEC then reopens the file in BACKUP mode. One page at a time, it scans the file line by line, calling the line number substitution macro to replace old line numbers with new ones. Then RESEQ.TEC scans for GOTO's, GOSUB's, etc., and makes the necessary substitutions for their arguments. When the entire file has been scanned, RESEQ.TEC exits, leaving the original file with a ".BAK" extension, and the newly renumbered version with the original name.

## PROGRAM INSTALLATION

1) If this program is to be run on a RSTS/E system, it should be compressed to reduce space and TRIPLE execution time. Since TECO is an interpreted language, it must

repeatedly scan program comments, spaces, etc., that are not integral to the program's execution. Removing all comments and most of the spaces (except for the space and tab in the first statement) will greatly improve program efficiency, although it will render the program highly unreadable. IT SHOULD BE NOTED THAT SOME SPACES ARE SIGNIFICANT WHEN BETWEEN TECO COMMANDS, AND THE PROGRAMMER WHO COMPRESSES THIS PROGRAM SHOULD BE THOROUGHLY FAMILIAR WITH THESE EXCEPTIONS.

2) The HELP file specification of [170,1]RESEQ.HLP must be altered if a help message is to be used.

3) The executable version of the program must have a ".TEC" extension, and must have the 64 (decimal) bit set in the protection code to indicate EXECUTABLE IMAGE.

RESEQ.TEC is a program which will take a Basic-Plus (or BASIC-PLUS II) file, and renumber the lines with uniform intervals between line numbers. As it renumbers the lines, RESEQ.TEC also changes the following statements to conforms to the new line sequence:

GOTO ln
ON...GOTO ln1,ln2,...
GOSUB ln
ON...GOSUB ln1,ln2,...
RESUME ln
IF...THEN ln
IF...THEN...ELSE ln
ERL (<,=,>,etc.) ln
LINE (<,=,>,etc.) ln

DIRECTIONS: RESEQ.TEC is run by typing RUN (170,1) RESEQ. RESEQ.TEC will respond with a header and the prompt:

TYPE '?' FOR HELP
FILE:

You then type in the name of the program you wish to resequence. If you type a question mark ('?'), this help message will appear. Typing the name of a non-existent file produces the error:

filename/B+ could not be found — please try again

(The '/B+' which appears after the file name is an operating feature of TECO, and can be ignored.)

Next, you will see the prompt:

OLD STARTING LINE NUMBER OF PROGRAM SEGMENT <1> ?

RESEQ.TEC is now asking you for the first line number of the program segment you wish to resequence. In most cases, you would want to start at the first line of the entire program, or line 1. Hitting a blank <RETURN> defaults this answer to 1.

The next prompt is:

OLD ENDING LINE NUMBER OF SEGMENT <32767> ?

RESEQ.TEC is asking you for the highest line number of the program segment you wish to resequence. In most cases, you would want to renumber to the end of the program, or line 32767, the greatest possible line number. Hitting a blank <RETURN> defaults this answer to 32767.

Now RESEQ.TEC will ask for:

ENTER NEW STARTING LINE NUMBER FOR SEGMENT <10> ?

RESEQ.TEC wants to know what the first line number of the NEW segment is to be (i.e., if you wanted to renumber your program so that the first line becomes line number 100, you'd answer 100 for this prompt). This answer defaults to 10 for a blank <RETURN>.

Finally, RESEQ.TEC asks:

ENTER THE INCREMENT FOR THE NEW LINE NUMBERS <10> ?

RESEQ.TEC now wants the increment for the new line numbers (i.e., if you wanted your program lines to become 100, 120, 140, etc., you'd specify an increment of 20). This answer defaults to 10 for a blank <RETURN>.

RESEQ.TEC will then print the message 'PASS 1'. RESEQ.TEC is looking at your program, and assembling a new line number table for the segment you wished to change. If by some chance the new line numbers overflow, or become greater than 32767 (the greatest possible line number), then RESEQ.TEC will print:

NEW LINE NUMBERS OVERFLOWED — RE-SPECIFY SEGMENT DIMENSIONS

and RESEQ.TEC will exit. Your original program will still be preserved.

If all goes well, RESEQ.TEC will then print the message 'PASS 2'. RESEQ.TEC is now substituting the new line numbers for the old ones in your program. Here a special problem may arise: if RESEQ.TEC finds a GOTO or other line reference to a non-existent line number, it will print the following message:

LINE NUMBER %%%%% NOT FOUND AT NEW LINE #####

where %%%%% represents the non-existent line number referenced, and ##### represents the new line number where the error can be found. RESEQ.TEC substitutes three question marks ('???') for the erroneous reference in the new program.

When RESEQ.TEC is done processing your program, the 'Ready' prompt (for the BASIC run-time system) will appear on your terminal. Your new program will be saved, and your old program will be preserved under a '.BAK' extension.

POSSIBLE PROBLEM: If your program is exceptionally large, RESEQ.TEC may not be able to handle it all at once. This problem can be EASILY FIXED by inserting form-feed (ASCII code 12) characters at various points in your program, splitting the program up into managable portions for RESEQ.TEC.

HAPPY RESEQUENCING!

```
!*************************************************!
!**                                             **!
!**                RESEQ.TEC V01                **!
!**             Lawrence P. Gallagher           **!
!**                  07-Apr-82                  **!
!**                                             **!
!**    LINE NUMBER RESEQUENCER FOR BASIC-PLUS   **!
!**    AND BASIC-PLUS II SOURCE FILES           **!
!**                                             **!
!**    MODIFIES ALL EMBEDDED LINE REFERENCES    **!
!**    (GOTO'S, ERL'S, ETC.)                    **!
!**                                             **!
!*************************************************!
```

```
! LOAD MACROS AND DATA !
! !
! !
^UD      <>=$                    !LOAD QD WITH A SPACE, TAB, <, =, > !
! !
```

```
@^UB#                            !LOAD QB WITH THE 'TERMINAL DRIVER' MACRO !
! !
!BEGIN TTY INPUT MACRO!
^D0U1
<  ^TU0                          !   GET A CHARACTER FROM TTY!
   (Q0-13)*(Q0-10)*(Q0-27)
         "E0EXIT$'               !   EXIT THIS MACRO IF DELIMITER TYPED !
   (Q0-3)"E ^C$'                 !   HALT IF CTRL/C TYPED !
   (Q0-18)"E13^T10^T.-Q1,.T
         ONXTCHR$'               !   RE-DISPLAY INPUT BUFFER IF CRTL/R  !
   (Q0-21)"E-Q1D0U17^T13^T10^T
         ONXTCHR$'               !   ZERO BUFFER IF CTRL/U  !
   (Q0-127)"EQ1"G-D-1$1$
         8^T32^T8^TONXTCHR$'' !  !   BACKSPACE AND ERASE ONE CHARACTER ON DEL !
                                 !   (THIS IS FOR SCOPE USAGE ONLY) !
!HERE WE ADD THE CHARACTER!
  Q0I$$1
!NXTCHR!>$
!EXIT!
(Q0-13)"E^TU0'                   !DUMP THE TRAILING LINE FEED ON <RET>!
(Q0-10)"E13^T'                   !PRINT THE NEEDED <CR> IF IT'S A LINE FEED !
.-Q1,.X0                         !LOAD Q0 WITH THE INPUT STRING!
.-Q1,.D                          !AND CLEAN UP THE BUFFER!
#
! !
!LOAD SUBSTITUTION MACRO!
@^UR#
<::S^EGD$;>                      !ADVANCE UNTIL CHR DOESN'T MATCH ANY!
                                 !      OF THOSE STORED IN QD!
.U1                              !STORE BUFFER POINTER IN Q1!
\U2                              !STORE THIS LINE NUMBER (IF ANY) IN Q2!
(Q2-QA+1)"G(QZ-Q2+1)"G           !IF LOWER LIMIT <= Q2 <= UPPER LIMIT...!
   ^U0.$                         !INSERT A '.'!
   13I$10I$                      !INSERT A CR/LF!
   Q1J                           !POSITION POINTER TO BEGINNING OF NUMBER!
   1:X0                          !AND APPEND THIS NUMBER INTO Q0!
   \U2                           !(THIS MOVES POINTER TO END OF NUMBER)!
   2D                            !DELETE THE ADDED CR/LF!
   :^U0.$                        !APPEND ANOTHER '.'!
   JQE,1:S^EQ0$U7                !DO A COLON-SEARCH IN THE TABLE!
   Q7"E ^ALINE NUMBER ^A         !IF THE NUMBER WASN'T FOUND...!
      Q2:=                       !     THEN THIS REFERENCE IS TO A!
      ^A NOT FOUND AT NEW LINE ^A !  NON-EXISTENT NUMBER!
      (Q1+:Q0-4)J                !  RE-POSITION POINTER!
      0L                         !MOVE TO BEGINNING OF LINE!
      <\U0                       !GRAB THE ERRONEOUS NUMBER!
      Q0-1;-L>Q0=                !AND PRINT IT WHEN FOUND!
      (Q1+:Q0-4)J                !RE-POSITION POINTER!
      Q1,.D                      !DELETE THE NUMBER!
      I???$'                     !SUBSTITUTE ??? FOR THE ERRONEOUS NUMBER!
!REPLACE!
   Q7"N\U3                       !READ THE NEW NUMBER FROM THE TABLE!
   (Q1+:Q0-4)J                   !RE-POSITION THE POINTER!
   Q1,.D                         !DELETE THE OLD NUMBER!
   Q3\'''                        !AND PLUG IN THE NEW ONE!
<::S^EGD$;>                      !PROCEED UNTIL CHR DOESN'T MATCH ANY IN QD!
(0A-44)"E                        !AND IF IT'S A COMMA (SUCH AS ON...GOTO)!
   C MR'                         !SET THE POINTER, AND CALL AGAIN RECURSIVELY!
#
                          !RESEQ.TEC  TOP END !

^ARESEQ.TEC    V01^A
2<13^T10^T>
HK                               !CLEAR BUFFER!
2ED                              !ENABLE YANKING!
<<  ^ATYPE '?' FOR HELP ^A13^T10^T
    ^AFILE: ^A
    HKMBG013^T10^T               !GET FILE SPEC AND STORE IN BUFFER!
    J:S?$;                  !EXIT THIS LOOP IF NOT A '?'!
:ER[170,1]RESEQ.HLP$U7           !OPEN HELP FILE (CHANGE THIS SPEC)!
    Q7"NYHTHK'                   !AND PRINT THE HELP TEXT (IF ANY)!
    Q7"E^ASorry, Help not available!^A2<13^T10^T>'
    >
    Z"E^APLEASE ENTER A FILE NAME^A2<13^T10^T>'
    Z"N:S.$U7                    !SEE IF HE SPECIFIED AN EXTENSION!
    Q7"EZJI.BAS$'                !NO, DEFAULT TO '.BAS'!
    ZJI/B+$                      !ADD THE '/B+' SWITCH!
    ^US:ER$                      !LOAD AN ':ER' INTO QS!
    ZJ0:XS                       !APPEND THE FILE NAME!
    27:^US$                      !APPEND AN 'ESCAPE'!
    HK                           !ZERO THE BUFFER!
    MSU0                         !ATTEMPT TO OPEN AND SAVE STATUS!
    (-Q0-1);                     !EXIF IF Q0 = -1 (AND REPEAT IF 0) !
    ^A? ^A:G*^A could not be found -- please try again^A  !PRINT ERROR!
    2<13^T10^T>'>                 !AND REPEAT!


<< ^AOLD STARTING LINE NUMBER OF PROGRAM SEGMENT <1> ?^A
   MBG0                          !PUT IN BUFFER!
   J\UAHK                        !SAVE IN QA AND CLEAR BUFFER!
   QA"E1UA'                      !DEFAULT TO (1)!
   QA;                           !AND EXIT IF VALUE IS POSITIVE!
   ^ATHE NUMBER MUST BE BETWEEN 1 AND 32767^A13^T10^T
 >

<^AOLD ENDING LINE NUMBER OF SEGMENT <32767> ?^A
   MBG0                          !PUT IN BUFFER!
   J\UZHK                        !SAVE IN QZ AND CLEAR BUFFER!
   QZ"E32767UZ'                  !DEFAULT TO 32767!
   QZ;                           !AND EXIT IF VALUE IS POSITIVE!
   ^AENDING NUMBER MUST BE BETWEEN 1 AND 32767^A13^T10^T
>
  QZ-QA;                         !MAKE SURE QZ>QA!
  13^T10^T
  ^AENDING LINE (^A QZ:= ^A) MUST BE GREATER THAN OR EQUAL TO ^A
  ^ASTARTING LINE (^A QA:= ^A) ^A2<13^T10^T>
>
13^T10^T
```

```
<^AENTER NEW STARTING LINE NUMBER FOR SEGMENT <10> ?^A
   MBG0                          !PUT IN BUFFER!
   J\ULHK                        !SAVE IN QL AND CLEAR BUFFER!
   QL"E10UL'                     !DEFAULT TO 10!
   QL;                           !EXIT IF POSITIVE!
   ^ASTARTING NUMBER MUST BE BETWEEN 1 AND 32767^A13^T10^T
>
<^AENTER THE INCREMENT FOR THE NEW LINE NUMBERS <10> ?^A
   MBG0                          !PUT IN BUFFER!
   J\UIHK                        !SAVE IN QI AND ZERO BUFFER!
   QI"E10UI'                     !DEFAULT TO 10!
   QI;                           !EXIT IF POSITIVE!
   ^AINCREMENT MUST BE BETWEEN 1 AND 32767^A13^T10^T
>
13^T10^T

!** PASS ONE **  ASSEMBLE THE LINE NUMBER SUBSTITUTION TABLE !

^APASS ONE^A 13^T 10^T
OUC                              !ZERO COUNTER!
Y                                !YANK FIRST PAGE!
<<\UT                            !STORE THIS LINE NUMBER (IF ANY) IN QT!
   (QT-QA+1)"G(QZ-QT+1)"G        !IF IN RANGE...!
   .UOQL\Q0,.XTQ0,.D             !STORE THE NEW LINE NO.....!
   [T %C
   (QI)%L                        !PUSH AND INCREMENT QC AND QL!
   QL"L^ANEW LINE NUMBERS OVERFLOWED -- RE-SPECIFY SEGMENT DIMENSIONS^A
       13^T10^T OVERYEND$'''     !IF OUR NEW LINE NUMBERS BECOME!
                                 !GREATER THAN 32767, WARN THE USER!
   L                             !NEXT LINE!
   (.-Z);                        !EXIT IF POINTER AT END OF PAGE!
>
Y                                !GET NEW PAGE!
-Z;                              !EXIT IF PAGE IS EMPTY (WE'RE DONE)!
>



!LOAD TABLE INTO QX!
QC<]T                            !POP FROM STACK INTO QT!
   I.$                           !INSERT A '.'!
   QT\                           !INSERT THE OLD NUMBER!
   13I$10I$                      !INSERT A CR/LF!
   I.$                           !INSERT A '.'!
   GT                            !INSERT THE NEW NUMBER!
  I.$
   13I$10I$                      !INSERT A NEW CR/LF!
  >
JQC*2XX                          !STORE THIS TABLE IN QX!
ZUE                              !STORE NO. CHRS IN QE!
HK                               !CLEAR BUFFER!
!** PASS TWO ** SEARCH FOR KEY WORDS AND SUBSTITUTE NUMBERS !

^APASS TWO^A 13^T 10^T
^USEB$                           !LOAD AN EB COMMAND!
G*0:XS                           !APPEND THE FILE SPEC!
27:^US$                          !APPEND THE 'ESC'!
HK                               !CLEAR THE BUFFER!
MS                               !OPEN THE FILE FOR INPUT!
Y                                !GET FIRST PAGE!
<
!RETRIEVE TABLE!
JGX                              !INSERT TABLE IN FRONT OF BUFFER!
0L
<\"N 0L                          !IF THIS IS A REAL LINE NO, PROCEED!
(0A-43)*(0A-45)"N                !IF THIS LINE NO IS IN RANGE...!
   MR''                          !THEN CALL THE SUBSTITUTION FUNCTION!
   L                             !ADVANCE A LINE!
   .-Z;                          !AND EXIT LOOP IF DONE!
>
QEJ:SGOTO$U7                     !FIND A GOTO!
   Q7"N<MR                       !REPLACE IT,!
   :SGOTO$;>'                    !AND TRY AGAIN!
QEJ:SGOSUB$U7                    !FIND A GOSUB...!
Q7"N<MR:SGOSUB$;>'
QEJ:STHEN$U7                     !FIND A 'THEN'...!
Q7"N<MR:STHEN$;>'
QEJ:SELSE$U7                     !FIND AN 'ELSE'...!
Q7"N<MR:SELSE$;>'
QEJ:SRESUME$U7                   !FIND A 'RESUME' ...!
Q7"N<MR:SRESUME$;>'
QEJ:SERL$U7                      !FIND AN 'ERL <' ,'ERL =', etc. !
Q7"N<MR:SERL$;>'
QEJ:SLINE$U7                     !FIND A 'LINE < ', 'LINE = ', etc. !
Q7"N<MR:SLINE$;>'

!NEXT PAGE!
JQED                             !CLEAR TABLE!
P                                !GRAB THE NEXT PAGE!
-Z;>                             !AND EXIT (WE'RE DONE) IF PAGE IS BLANK!
EX                               !BYE BYE !
!VERYEND!
^C$$
```

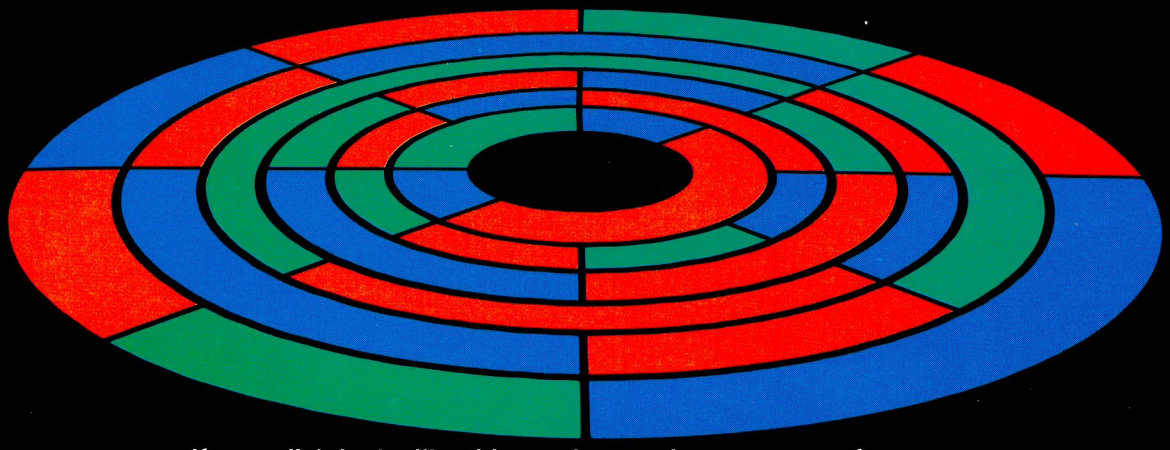
♥

# RTS: Test Runtime System Example

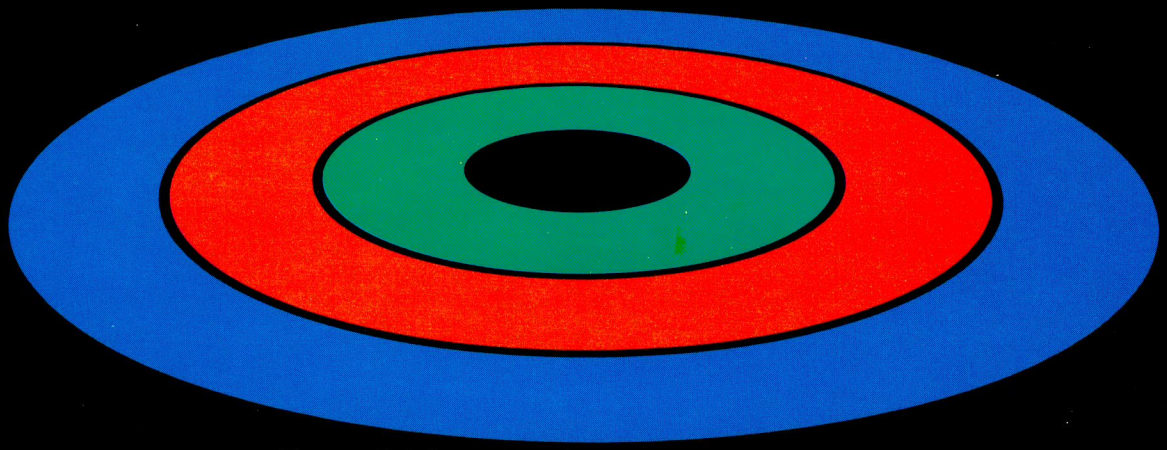
. . . continued from page 55

```
 86
 87 000342                                      PUSH    R5                      ;SAVE STRING LENGTH
 88 000344  010402                              MOV     R4,R2                   ;GET BUFFER ADDR
 89 000346  111203                   CVTLOP::MOVB    @R2,R3                  ;GET A CHAR
 90 000350  120327  000173                      CMPB    R3,#173                 ;CHECK FOR UPPER/LOWER
 91 000354  103006                              BHIS    SOBLOP                  ;ABOVE 'z'
 92 000356  120327  000140                      CMPB    R3,#140
 93 000362  101403                              BLOS    SOBLOP                  ;BELOW 'a'
 94 000364  162703  000040                      SUB     #32.,R3                 ;CONVERT TO UPPER
 95 000370  110312                              MOVB    R3,@R2                  ;PUT IT BACK
 96 000372  005202                   SOBLOP::INC     R2
 97 000374  077514                              SOB     R5,CVTLOP               ;DO UNTIL DONE
 98 000376                                      POP     R5                      ;RESTORE LENGTH
 99
100                                 ; check for RTS command
101
102 000400  010402                              MOV     R4,R2                   ;SETUP FOR PARSE COMMAND LOOP
103 000402                                      CALL    PRSCMD                  ;PROCESS THE COMMAND
104 000406  005700                              TST     R0                      ;COULD WE PROCESS IT?
105 000410  001051                              BNE     DOCMD                   ;YES, LETS GO DO IT
106
107                                 ; check for CCL command
108
109 000412                                      CLRXRB                          ;CLEAR XRB STUFF
110 000416  010567  000442                      MOV     R5,XRB+XRLEN            ;SETUP AS READ RETURNED
111 000422  016767  000442  000444              MOV     XRB+XRLEN,XRB+XRBC
112 000430  010467  000446                      MOV     R4,XRB+XRLOC
113 000434  104062                              .CCL                            ;DO WE HAVE A CCL???
114
115                                 ; not a valid anything if we get here
116
117 000436  116702  000402                      MOVB    FIRQB,R2                ;GET ERROR THAT OCCURED
118 000442  105702                              TSTB    R2                      ;ANYTHING RCVD???
119 000444  001415                              BEQ     BADP                    ;NO, SO USE STD MESSAGE
120
121 000446                                      MESSAGE #BDSMSG,#17.            ;TELL USER
122 000476  000414                              BR      BDRST
123
124 000500                          BADP:   MESSAGE #BADCMD,#8.
125 000530  000167  177360          BDRST:  JMP RTSINP                          ;NOPE, SORRY, LETS PRCESS THIS
126
127                                 ; process a  RTS command
128
129 000534  006303                  DOCMD:: ASL     R3                      ;GENERATE OFFSET FOR FOLLOWING TABLE
130 000536  012704  000544'                  MOV   - #DISPAT-2,R4            ;GET DISPATCH ADDR
131 000542  060304                          ADD     R3,R4                   ;OFFSET IT
132 000544  011407                          MOV     @R4,PC                  ;DO THE JMP
133
134 000546  000564'                 DISPAT::.WORD   OUTCMD                  ;'OUT'
135 000550  000576'                         .WORD   HLPCMD                  ;'HELP'
136 000552  000662'                         .WORD   VERCMD                  ;'VERSION'
137 000554  000746'                         .WORD   RUNCMD                  ;'RUN'
138 000556  001170'                         .WORD   ASSCMD                  ;'ASSIGN'
139 000560  001046'                         .WORD   DEACMD                  ;'DEASSIGN'
140 000562  000000                          .WORD   0                       ;END OF DISPATCH
141
142 000564                          OUTCMD::CLRXRB
143 000570  104046                          .EXIT                           ;BACK TO NORMAL RTS
144 000572  000167  177316                  JMP     RTSINP                  ;NEVER GETS HERE
145
146 000576                          HLPCMD::MESSAGE #RTSNAM,RTSLEN          ;PRINT RTS NAME
147 000626                                  MESSAGE #VERMSG,#231.           ;LETS GET HELP
148 000656  000167  177232                  JMP     RTSINP                  ;BACK TO USER
149
150 000662                          VERCMD::MESSAGE #RTSNAM,RTSLEN          ;PRINT RTS NAME
151 000712                                  MESSAGE #VERMSG,#13.            ;TELL USER RTS VERSION
152 000742  000167  177146                  JMP     RTSINP                  ;BACK TO USER
153
154 000746                          RUNCMD::CLRXRB                          ;CLEAR IT
155 000752                                  CLRFQB                          ;DITTO
156
157 000756  010567  000442                  MOV     R5,XRB+XRLEN            ;ABOUT RIGHT
158 000762  010567  000444                  MOV     R5,XRB+XRBC             ;DITTO
159 000766  010267  000446                  MOV     R2,XRB+XRLOC            ;WHERE ARG BEGINS
160 000772  104064                          .FSS                            ;GET FILENAME
161 000774                                  TSTFQB                          ;ANY ERRORS???
162
163 001012  005767  000416          OKFILE::TST     FIRQB+FQEXT             ;ANY EXTENSION???
```

# WHAT YOU DON'T KNOW ABOUT YOUR DISKS IS COSTING YOU MONEY



**If your disk looks like this, you're wasting system performance.**



**If your disk looks like this, you're using DISKIT.**

When the job you're running requires reading the "red" file, it naturally happens faster on a well-ordered disk. Disks become "fragmented" as you use your computer. The system slows down. And that costs you money.

Now, you can restructure your disks and get back that lost performance (up to 50%) without spending a dime on new hardware. DISKIT is the original software system that makes this possible.

But don't confuse DISKIT with other system utilities, DISKIT is a complete "software tool kit" that optimizes your RSTS/E system.

DISKIT is:

- DSU — The utility which restructures the information on your disk, making data fast and easy to access.
- DIR — The incredible directory tool that finds files at the rate of 400 per second.
- RDR — Reorders disk directories 30 times faster than ever before possible.
- OPEN — Displays complete job statistics and file activity so you can see what your system is doing.
- DUS — The set of CALLable subroutines which pre-extend file directories, reducing fragmentation.

In today's tight economy, it's more important than ever to get the most out of your hardware investment. Call or write today and start getting your money's worth from your computer.

Software Techniques Incorporated

5242 Katella Avenue
Los Alamitos, CA 90720
United States
Phone: [714] 995-0533

287 London Road
Newbury, Berkshire RG13 2QJ
United Kingdom
Phone: 44 [0] 635-30840

CIRCLE 65 ON READER CARD

```
164 001016  001003                         BNE     NOTWLD                  ;YES
165
166 001020  012767  177777  000416         MOV     #-1,FIRQB+FQEXT         ;SET WILDCARD EXTENSION
167
168 001026  104042                  NOTWLD::.RUN                            ;TRY TO DO IT
169
170 001030                                  TSTFQB                          ;ANY ERRORS???
171
172                                 ; deassign command
173
174 001046  005702                  DEACMD::TST     R2                      ;ANY ARGS???
175 001050  001021                         BNE     DESONE                  ;YES, DO CHECK WHAT TO DEASSIGN
176
177 001052                                  CLRFQB
178 001056                                  CLRXRB
179 001062  112767  000014  000405         MOVB    #UU.DAL,FIRQB+FQFUN     ;SET UP DEASSIGN
180 001070  104076                         .ULOG                           ;CLEAR ALL
181 001072                                  TSTFQB                          ;PRINT ANY ERROR ENCOUNTERED
182 001110  000167  177000                 JMP     RTSINP                  ;DONE
183
184 001114                          DESONE::CLRXRB                          ;CLEAR IT
185 001120                                  CLRFQB                          ;DITTO
186
187 001124  010567  000442                 MOV     R5,XRB+XRLEN            ;ABOUT RIGHT
188 001130  010567  000444                 MOV     R5,XRB+XRBC             ;DITTO
189 001134  010267  000446                 MOV     R2,XRB+XRLOC            ;WHERE ARG BEGINS
190 001140  104064                         .FSS                            ;GET FILENAME
191 001142                                  TSTFQB                          ;ANY ERRORS???
192
193 001160                          OKDEAS:                                 ;NO, SO LETS DO IT
194 001160  012705  000013                 MOV     #UU.DEA,R5              ;SETUP DEASSIGN ,JUMP TO ASSIGN LOGIC
195 001164  000167  000050                 JMP     ASSNGO                  ;DO IT PLEASE
196
197                                 ; assign command
198
199 001170                          ASSCMD::CLRXRB                          ;CLEAR IT
200 001174                                  CLRFQB                          ;DITTO
201
202 001200  010567  000442                 MOV     R5,XRB+XRLEN            ;ABOUT RIGHT
203 001204  010567  000444                 MOV     R5,XRB+XRBC             ;DITTO
204 001210  010267  000446                 MOV     R2,XRB+XRLOC            ;WHERE ARG BEGINS
205 001214  104064                         .FSS                            ;GET FILENAME
206 001216                                  TSTFQB                          ;ANY ERRORS???
207
208 001234  012705  000012          OKASSG: MOV     #UU.ASS,R5              ;SETUP ASSIGN, DO W/CODE SET
209 001240                          ASSNGO:                                 ;R5 = 13=DEASSIGN,12=ASSIGN
210
211 001240  032767  100000  000452         BIT     #100000,XRB+10          ;GET BASIC FLAG-WORD 2
212 001246  001410                         BEQ     OKASGN                  ;LEGAL DEVICE
213
214 001250                                  ERROR   #NODEVC                 ;NOT A VALID DEVICE
215 001264  000167  176624                 JMP     RTSINP                  ;GET NEW PROMPT
216
217 001270  110567  000405          OKASGN: MOVB    R5,FIRQB+FQFUN          ;ASSIGN OR DEASSIGN CODE
218 001274                                  CLRXRB
219 001300  104076                         .ULOG
220 001302                                  TSTFQB                          ;SEE IF ERROR
221 001320  000167  176570                 JMP     RTSINP
222
224
225                                 ; asynchronous traps
226
227 001324  104026                  CCTRAP::.TTRST                          ;SYS(CHR$(0%))
228 001326  000167  176562                 JMP     RTSINP                  ;IGNORE CTRL/C, MUST USE 'OUT'
229
230 001332                          BDERRO::                                ;ANY WEIRD ERRORS
231 001332                                  MESSAGE #FTLERR,#28.
232 001362  104046                         .EXIT
233
235
236                                 ; some useful subroutines
237
238 001364                          $FRBCL:: PUSH   R1                      ;SAVE R1
239 001366  012701  000402                  MOV    #FIRQB, R1              ;START OF FIRQB
240 001372  005021                          CLR    (R1)+
241 001374  005021                          CLR    (R1)+
242 001376  005021                          CLR    (R1)+
243 001400  005021                          CLR    (R1)+
244 001402  005021                          CLR    (R1)+
245 001404  005021                          CLR    (R1)+
246 001406  005021                          CLR    (R1)+
247 001410  005021                          CLR    (R1)+
```

```
248 001412  005021                         CLR     (R1)+
249 001414  005021                         CLR     (R1)+
250 001416  005021                         CLR     (R1)+
251 001420  005021                         CLR     (R1)+
252 001422  005021                         CLR     (R1)+
253 001424  005021                         CLR     (R1)+
254 001426  005021                         CLR     (R1)+
255 001430  005021                         CLR     (R1)+
256 001432                                 POP     R1
257 001434                                 RETURN
258
259 001436                          $XRBCL:: PUSH   R1                              ;SAVE R1
260 001440  012701  000442                 MOV     #XRB, R1
261 001444  005021                         CLR     (R1)+
262 001446  005021                         CLR     (R1)+
263 001450  005021                         CLR     (R1)+
264 001452  005021                         CLR     (R1)+
265 001454  005021                         CLR     (R1)+
266 001456  005021                         CLR     (R1)+
267 001460  005021                         CLR     (R1)+
268 001462                                 POP     R1                              ;GET IT BACK
269 001464                                 RETURN
270
271 001466                          $ERRPT::
272 001466                                 CLRFQB
273 001472  112767  000011  000405         MOVB    #UU.ERR,FIRQB+FQFUN
274 001500  110167  000406                 MOVB    R1,FIRQB+FQERNO
275 001504  104066                         .UUO
276 001506                                 MESSAGE #FIRQB+4,#28.
277 001536                                 MESSAGE #CRLF,#2.
278 001566                                 RETURN
279
280 001570                          $PRFRQ:: PUSH   R3
281 001572  116703  000402                 MOVB    FIRQB,R3
282 001576                                 ERROR   R3                              ;PRINT AN ERROR MESSAGE
283 001610                                 POP     R3
284 001612                                 RETURN
285
286 001614                          PRSCMD::
287                                 ;       CHECKS FOR VALID COMMAND FOR THIS RUN TIME SYSTEM
288                                 ;       IF IT IS, IT SETS R0 <>0, ELSE R0=0
289                                 ;
290                                 ;       INPUTS: R2 = COMMAND TO COMPARE TO TABLE
291                                 ;       OUTPUT: R3 = INDEX OF COMMAND NUMBER TO EXECUTE
292                                 ;
293                                 ;       NOTE: R4 IS DESTROYED
294                                 ;
295 001614  005000                         CLR     R0              ;ASSUME WE CAN'T
296 001616  012701  002436'                MOV     #CMDTBL,R1      ;GET COMMAND TABLE ADDRESS
297 001622  012703  000001                 MOV     #1, R3          ;FIRST INDEX IN TABLE
298 001626  010204                         MOV     R2,R4           ;SAVE USER COMMAND BUFFER
299
300 001630  122221                  PRSLOP: CMPB    (R2)+,(R1)+     ;LETS DO A CHARACTER BY CHARACTER SCAN
301 001632  001020                         BNE     SKIPCM          ;NO MATCH, LETS SKIP TO NEXT CMDTBL
302
303 001634  105711                         TSTB    @R1             ;INCREMENTED END OF COMMAND IN TBL?
304 001636  001374                         BNE     PRSLOP          ;NOPE, CHECK MORE OF INPUT
305
306 001640  121227  000040                 CMPB    @R2,#32.        ;USER HAS A SPACE?
307 001644  001002                         BNE     NOARG           ;NOPE, LETS CHECK FOR TERMINATOR
308
309 001646  005202                         INC     R2              ;POINT PAST SPACE
310 001650  000421                         BR      OKWARG          ;SAY SUCCESSFUL, POINTING TO ARG...
```

```
311
312 001652  121227  000012          NOARG:: CMPB    @R2,#10.        ;USER ENTERED TERMINATOR??? (LF)
313 001656  001415                          BEQ     OKNOAR          ;YES, SAY SUCCESSFUL
314
315 001660  121227  000015                  CMPB    @R2,#13.        ;USER ENTERED TERMINATOR??? (CR)
316 001664  001412                          BEQ     OKNOAR          ;YES, SAY SUCCESSFUL
317
318 001666  121227  000033                  CMPB    @R2,#27.        ;USER ENTERED TERMINATOR??? (ESC)
319 001672  001407                          BEQ     OKNOAR          ;YES, SAY SUCCESSFUL
320
321 001674  105721                  SKIPCM: TSTB    (R1)+           ;GET TO END OF CURRENT CMD
322 001676  001376                          BNE     SKIPCM
323
324 001700  105711                          TSTB    @R1             ;END OF CMD REACHED, LETS CHECK END OF TBL
325 001702  001406                          BEQ     BDCMD           ;YEP, MUST BE BAD USER CMD
326
327 001704  005203                          INC     R3              ;INC INDEX, MORE CMDS TO CHECK
328 001706  010402                          MOV     R4,R2           ;RESTORE USER BUFFER ADDRESS
329 001710  000747                          BR      PRSLOP          ;...AND CHECK MORE
330
331 001712  005002                  OKNOAR::CLR     R2              ;SIGNAL NO ARGUMENTS
332 001714  012700  000001          OKWARG::MOV     #1,R0           ;SIGNAL SUCCESSFUL
333 001720                          BDCMD::RETURN                   ;BACK TO PROCESSING
334                                                                 ;R0 = 0,BAD CMD   <>0 = GOOD CMD
335                                                                 ;R2 = 0, NO ARG   <>0 = ADDR OF ARG FOUND
336                                                                 ;R3 = <>0 = INDEX OF COMMAND RECIEVED
337
339
340                                 ; messages
341
342                                         .ENABL LC
343 001722     015     012     117  PROMPT::.ASCIZ <15><12>/Ok/<15><12><12>
    001725     153     015     012
    001730     012     000
344 001732     040     040     040          .ASCIZ /         /                      ;EXTRA PROMPT PATCH SPACE
    001735     040     040     040
    001740     040     040     040
    001743     000
345 001744     015     012     102  BYEMSG::.ASCIZ <15><12>/Bye/<15><12><12>
    001747     171     145     015
    001752     012     012     000
346                                         .EVEN
347 001756  054746                  PRGNAM::.RAD50  /NON/                          ;ALLOWS NONAME PATCHING
348 001760  004115                          .RAD50  /AME/
349 001762     077     127     150  BADCMD::.ASCIZ  /?What?/<15><12>               ;bad command input
    001765     141     164     077
    001770     015     012     000
350 001773     077     125     156  FTLERR::.ASCIZ  /?Undefined error occured?/<15><12>
    001776     144     145     146
    002001     151     156     145
    002004     144     040     145
    002007     162     162     157
    002012     162     040     157
    002015     143     143     165
    002020     162     145     144
    002023     077     015     012
    002026     000
351 002027     077     111     154  BDSMSG::.ASCIZ  /?Illegal switch/<15><12>
    002032     154     145     147
    002035     141     154     040
    002040     163     167     151
    002043     164     143     150
    002046     015     012     000
352
353 002051     040     055     040  VERMSG::.ASCII  / - /                                    ;3
354 002054     126     060     061  VERSON::.ASCIZ  /V01.00.1/<15><12>                      ;10
    002057     056     060     060
    002062     056     061     015
    002065     012     000
355 002067     015     012     103  HLPMSG::.ASCII  <15><12>/Commands:/<15><12><12>         ;14
    002072     157     155     155
    002075     141     156     144
    002100     163     072     015
    002103     012     012
356 002105     122     125     116          .ASCII  /RUN     - Run a program/<15><12>       ;25
    002110     040     040     040
    002113     040     040     055
    002116     040     122     165
    002121     156     040     141
    002124     040     160     162
    002127     157     147     162
    002132     141     155     015
```

```
                002135  012
357             002136  117     125     124             .ASCII  /OUT     - Out to system default KBM/<15><12>   ;37
                002141  040     040     040
                002144  040     040     055
                002147  040     117     165
                002152  164     040     164
                002155  157     040     163
                002160  171     163     164
                002163  145     155     040
                002166  144     145     146
                002171  141     165     154
                002174  164     040     113
                002177  102     115     015
                002202  012
358             002203  126     105     122             .ASCII  /VERSION - Type version number of RTS/<15><12>  ;38
                002206  123     111     117
                002211  116     040     055
                002214  040     124     171
                002217  160     145     040
                002222  166     145     162
                002225  163     151     157
                002230  156     040     156
                002233  165     155     142
                002236  145     162     040
                002241  157     146     040
                002244  122     124     123
                002247  015     012
359             002251  101     123     123             .ASCII  /ASSIGN  - Assign devices or logicals/<15><12>  ;38
                002254  111     107     116
                002257  040     040     055
                002262  040     101     163
                002265  163     151     147
                002270  156     040     144
                002273  145     166     151
                002276  143     145     163
                002301  040     157     162
                002304  040     154     157
                002307  147     151     143
                002312  141     154     163
                002315  015     012
360             002317  104     105     101             .ASCII  /DEASSIGN- Deassign devices or logicals/<15><12> ;40
                002322  123     123     111
                002325  107     116     055
                002330  040     104     145
                002333  141     163     163
                002336  151     147     156
                002341  040     144     145
                002344  166     151     143
                002347  145     163     040
                002352  157     162     040
                002355  154     157     147
                002360  151     143     141
                002363  154     163     015
                002366  012
361             002367  110     105     114             .ASCIZ  /HELP    - This message/<15><12>               ;24
                002372  120     040     040
                002375  040     040     055
                002400  040     124     150
                002403  151     163     040
                002406  155     145     163
                002411  163     141     147
                002414  145     015     012
                002417  000
362             002420  015     012     000     CRLF::  .ASCIZ  <15><12>                                      ;2
363                                                     .EVEN
364
365             002424  122     124     123     RTSNAM::.ASCIZ  /RTS/                   ;RTSNAM - PATACHABLE
                002427  000
366             002430  000     000     000             .BYTE   0,0,0,0
                002433  000
367             002434  000003                  RTSLEN::.WORD   3                       ;LEN MESSAGE- PATCHALBE
368                                                     ;
369                                                     ;CMDTBL, RTS SPECIAL COMMANDS
370                                                     ;  EACH COMMAND WILL END WITH A NULL BYTE
371                                                     ;  THE TABLE WILL END WITH A NULL BYTE
372                                                     ;
373             002436  117     125     124     CMDTBL::.ASCIZ  /OUT/
                002441  000
374             002442  110     105     114             .ASCIZ  /HELP/
                002445  120     000
375             002447  126     105     122             .ASCIZ  /VERSION/
                002452  123     111     117
                002455  116     000
```

```
376 002457  122  125  116          .ASCIZ  /RUN/
    002462  000
377 002463  101  123  123          .ASCIZ  /ASSIGN/
    002466  111  107  116
    002471  000
378 002472  104  105  101          .ASCIZ  /DEASSIGN/
    002475  123  123  111
    002500  107  116  000
379 002503  000  000               .BYTE   0,0
380
381                                .EVEN
382                                .DSABL LC
383
385
386                          ;***************VECTORS AND MISC RTS STUFF**********
387
388                          ;EXTEND VECTOR AREA .99998 - USED BY TASK-BUILD, MAKSIL
389
390 000000                       .PSECT  .99998,RO,GBL
391
392 000000                       .PSECT  .99999,RO,GBL
393
394 000000                   VECT$$::
395 000000  000400               .WORD PF.KBM                ;P.FLAG -RTS FLAG WORDS
396 000002  100014               .RAD50 /TST/                ;P.DEXT -DEFAULT EXTENSION
397 000004  000000               .WORD 0                     ;P.ISIZ -MUST BE 0
398 000006  000001               .WORD 1                     ;P.MSIZ -MINIMUN USER AREA
399 000010  001332'              .WORD BDERRO                ;P.FIS  -FLOATING POINT
400 000012  001332'              .WORD BDERRO                ;P.CRAS -SYS CRASH ENTRY
401 000014  000000'              .WORD RTS                   ;P.STRT -PRIMARY RTS ENTRY
402 000016  000012'              .WORD RTSNEW                ;P.NEW  -SWITCH ENTRY
403 000020  000074'              .WORD RTSRUN                ;P.RUN  -RUN FILE ENTRY
404 000022  001332'              .WORD BDERRO                ;P.BAD  -TRAP VECTOR
405 000024  001332'              .WORD BDERRO                ;P.BPT  -BREAKPOINT ENTRY
406 000026  001332'              .WORD BDERRO                ;P.IOT  -IOT INSTRUCTION ENTRY
407 000030  001332'              .WORD BDERRO                ;P.EMT  -EMULATOR TRAP
408 000032  001332'              .WORD BDERRO                ;P.TRAP -TRAP INST ENTRY
409 000034  001332'              .WORD BDERRO                ;P.FPP  -FLOAT POINT ENTRY
410 000036  001324'              .WORD CCTRAP                ;P.CC   - 1 ^C ENTERED
411 000040  001324'              .WORD CCTRAP                ;P.2CC  - 2 OR MORE ^C ENTERED
412 000042  000034               .WORD 28.                   ;P.SIZE - MAX USER AREA SIZE
413
414         000000'              .END RTS
```

SYMBOL TABLE

```
ASSCMD  001170RG    002 FQBUFL  000020        PF.KBM  000400        UMPFQ   000012        UU.YLG  177777
ASSFQ   000024          FQCLUS  000034        PF.NER  004000        USRLOG  000740        UU.ZER  000015
ASSNGO  001240R     002 FQDEV   000030        PF.REM  010000        USRPPN  000734        VECT$$  000000RG    004
ATRFQ   000000          FQDEVN  000032        PF.RW   002000        USRPRT  000736        VERCMD  000662RG    002
AUDHND  000032          FQERNO  000004        PF.SLA  040000        USRSP   000400        VERMSG  002051RG    002
BADCMD  001762RG    002 FQEXT   000014        PF.1US  001000        UUOFQ   000014        VERSON  002054RG    002
BADP    000500R     002 FQFIL   000004        PKBHND  000020        UU.ACT  177761        XRB     000442
BDCMD   001720RG    002 FQFLAG  000024        PLTHND  000034        UU.ASS  000012        XRBC    000002
BDERRO  001332RG    002 FQFUN   000003        PRGNAM  001756RG    002 UU.ATR  177747        XRBLK   000010
BDRST   000530R     002 FQJOB   000002        PROMPT  001722RG    002 UU.ATT  000006        XRBLKM  000007
BDSMSG  002027RG    002 FQMODE  000022        PRSCMD  001614RG    002 UU.BCK  177765        XRBSIZ  000016
BUFFER= 001002          FQNAM1  000010        PRSLOP  001630R     002 UU.BYE  000005        XRCI    000006
BYEMSG  001744RG    002 FQNAM2  000020        PTPHND  000012        UU.CCL  177750        XRLEN   000000
CALFIP  104000          FQNENT  000036        PTRHND  000010        UU.CHE  000023        XRLOC   000004
CCTRAP  001324RG    002 FQPFLG  000026        P.BAD   177754        UU.CHU  000010        XRMOD   000014
CDRHND  000014          FQPPN   000006        P.BPT   177756        UU.CLN  000002        XRTIME  000012
CLSFQ   000000          FQPROT  000027        P.CC    177770        UU.CNV  000024        $ERRPT  001466RG    002
CMDTBL  002436RG    002 FQSIZ   000016        P.CRAS  177744        UU.DAL  000014        $FRBCL  001364RG    002
CORCMN  000460          FQSIZM  000005        P.DEXT  177734        UU.DAT  177762        $PRFRQ  001570RG    002
CORLOP  000164R     002 FTLERR  001773RG    002 P.EMT   177762        UU.DEA  000013        $XRBCL  001436RG    002
CRAFQ   000004          HLPCMD  000576RG    002 P.FIS   177742        UU.DET  000007        .CCL    104062
CRBFQ   000034          HLPMSG  002067RG    002 P.FLAG  177732        UU.DIE  177760        .CHAIN  104070
CREFQ   000004          IBMHND  000042        P.FPP   177766        UU.DIR  000017        .CLEAR  104056
CRLF    002420RG    002 JFBIG   020000        P.IOT   177760        UU.DLU  000001        .CORE   104006
CRTFQ   000032          JFFPP   001000        P.ISIZ  177736        UU.DMP  177745        .DATE   104034
CVTLOP  000346RG    002 JFLOCK  040000        P.MSIZ  177740        UU.ERR  000011        .ERLOG  104052
DALFQ   000030          JFNOPR  010000        P.NEW   177750        UU.FCB  177770        .EXIT   104046
DDNFS   000400          JFPRIV  002000        P.OFF   177732        UU.FIL  177746        .FSS    104064
DDRLO   001000          JFSPRI  000400        P.RUN   177752        UU.HNG  177767        .LOGS   104054
DDWLO   002000          JFSYS   004000        P.SIZE  177774        UU.JOB  000030        .MESAG  104060
DEACMD  001046RG    002 KEY     000400        P.STRT  177746        UU.KMC  000033        .NAME   104044
DEAFQ   000026          KMCHND  000040        P.TRAP  177764        UU.LIN  000004        .PEEK   104012
DESONE  001114RG    002 LOKFQ   000022        P.2CC   177772        UU.LOG  177755        .PLAS   104072
DIRFQ   000012          LPTHND  000006        RENFQ   000010        UU.LOK  000021        .POSTN  104032
DISPAT  000546RG    002 MAPFQ   000010        RJEHND  000024        UU.MNT  000003        .READ   104002
DLNFQ   000006          MBXHND  000044        RSTFQ   000020        UU.NAM  177757        .RSX    104074
DMCHND  000030          MTAHND  000016        RTS     000000RG    002 UU.NLG  177776        .RTS    104050
DMPHND  000046          NOARG   001652RG    002 RTSINP  000114RG    002 UU.NME  000034        .RUN    104042
```

```
DOCMD   000534RG    002 NODEVC= ****** GX       RTSLEN  002434RG    002 UU.PAS  000000        .SET    104036
DROPCH  000264RG    002 NOTWLD  001026RG    002 RTSNAM  002424RG    002 UU.POK  177772        .SLEEP  104010
DSKHND  000000          NSTORG  001000        RTSNEW  000012R     002 UU.PPN  000031        .SPEC   104014
DTAHND  000004          NULHND  000026        RTSNME  000060R     002 UU.PRI  177763        .STAT   104040
DTRFQ   000002          OKASGN  001270R   002 RTSRED  000150R     002 UU.RAD  000016        .TIME   104030
DT2HND  000036          OKASSG  001234R   002 RTSRUN  000074RG    002 UU.RTS  177756        .TTAPE  104016
ELAFQ   000006          OKDEAS  001160R   002 RUNCMD  000746RG    002 UU.SLN  000025        .TTDDT  104024
ERRFQ   000016          OKFILE  001012RG  002 RXDHND  000022        UU.SPL  177744        .TTECH  104020
FIRQB   000402          OKNOAR  001712RG  002 SKIPCM  001674R     002 UU.SWP  000027        .TTNCH  104022
FLGFRC  020000          OKREAD  000270RG  002 SOBLOP  000372RG    002 UU.SYS  000032        .TTRST  104026
FLGKB   040000          OKWARG  001714RG  002 SPCLOP  000250R     002 UU.TB1  177775        .ULOG   104076
FLGMOD  010000          OPNFQ   000002        SYSVEE  030456          UU.TB2  177764        .UUO    104066
FLGPOS  004000          OUTCMD  000564RG  002 SYSVEL  033460          UU.TB3  177743        .WRITE  104004
FLGRND  100000          PF.CSZ  020000        TTYHND  000002          UU.TRM  000020        .XPEEK  104100
FQBSIZ  000040          PF.EMT  100000

. ABS.  177777      000
        000000      001
RTSMON  002506      002
```

## AUTHOR! AUTHOR!

***The RSTS PROFESSIONAL wants you to be an author!***

The RSTS PROFESSIONAL is your magazine. You can make it beter by contributing articles, programs or comments directly to us. Our authors are paid honoraria for published works, which because of thier hard work, they deserve. We ask you to contribute; send us your manuscripts for possible publication (we prefer machine readable tapes or floppies in PIP, RNO, WORD-11, or ?? format) to: RSTS PROFESSIONAL, P.O. Box 361, Ft. Washington, PA 19034-0361, Attn: Editors. *Thank you.*

I don't even think about computers once I leave the office, because I change modes, which enables this component to interface with my family.

# NEW PRODUCTS

## Texprint Emulator Gives Graphics Compatibility

Burlington, MA — Full compatibility with Plot-10 and Regis (vector-oriented computer graphics software languages) is now available on the Digital Equipment Corporation VT-125 graphics display terminal with the Texprint 4014 Emulator. Graphics users can now have both Tektronix Plot-10 compatibility and DEC Regis compatibility in the same CRT unit, thereby simplifying system operation, and often eliminating the need for special graphic display terminals.

Texprint's 4014 Emulator module allows graphics programs previously completed using Plot-10 vector commands to display images directly on the bit-map oriented VT-125 terminal, eliminating the costly and time consuming step of reprograming for DEC-compatible use. Because the 4014 Emulator preserves all the powerful features of VT-125 full color dot-addressable graphics and text, and also allows operation in the popular VT-100 mode, the Texprint-enhanced VT-125 display terminal is like having three different terminals in one.

Computer graphics users now no longer need to make a choice between Tektronix-compatibility (Plot-10) and DEC-compatibility (Regis) for their display terminal requirements, since the 4014 Emulator VT-125 combination fits both needs, at nominal cost, with no performance penalty. Hard-copy output is conveniently available for Tektronix graphics users with the VT-125/4014 Emulator and any Texprint-enhanced LA-120 printer/plotter terminal, which may be connected directly to the VT-125 printer port for fast "screen dump" operation.

Installation of the plug-in 4014 Emulator module in a VT-125 display terminal takes only a few minutes, and no special tools or skills are required. Each Texprint kit includes complete operator instructions and a one-year warranty.

4014 Emulator kits for the VT-125 are available from Texprint at $795; delivery is from stock.

## Western Resource Offers Fault-Tolerant Systems

Anaheim, CA — A new family of DEC-based Fault Tolerant Computer Systems was introduced by Western Resource Technology, Inc. at the recent Second National DEC-Compatible Exposition.

The new systems, built around DEC Q-Bus and UNIBUS based computers, provide a hardware solution to the problem of fault detection, tolerance and resolution. These systems support the standard DEC operating systems and will require little, if any, unusual programming efforts on the part of the user. People most interested in these products initially will be current DEC users that need to handle large numbers of devices or users, and can't suffer even minimum system down time. Technical systems developers, batch and process control groups, and commercial transaction processing/networking organizations are expected to be in this group.

"We looked around at the commercial systems providing high reliability or fault tolerance, and found nothing really profound," explained John Sutherland, WRT president. "Combining that finding with the fact that none of the suppliers were working with DEC technology, we decided to build DEC hardware based, fault responsive systems. In that way, we could easily beat the reaction speed and software sophistication problems, and appeal to the extensive DEC customer base already established.

"Every fault tolerant solution involves redundancy. When you have multiple systems, communications between them is the first problem to deal with. The next problem has to do with modularity. The system network must be able to communicate rapidly, unhook any detected faulty components, and keep the rest of the network running smoothly. Incidently, when you can do all of these things, you can also now overtly take down portions of a system for repair, preventive maintenance, or configuration change. This is a nice capability," he concluded.

The key to the WRT FTS Systems is the company's three new hardware subsystems; the Memory Duplication System (MDS), the Bus Isolation System (BIS), and the Power Control Subsystem (PCS).

"The MDS is the fastest DEC oriented memory-to-memory communications device I've ever seen. It moves a word or byte from one machine to another at literally bus speeds," he said. "The MDS, as with all of our current products, works on either the 18 or 22-bit Q-Bus, and the UNIBUS. The device's window can range from 2KB to 128KB in both directions. It also has eight interrupt lines in each direction, the last of which is a watch dog."

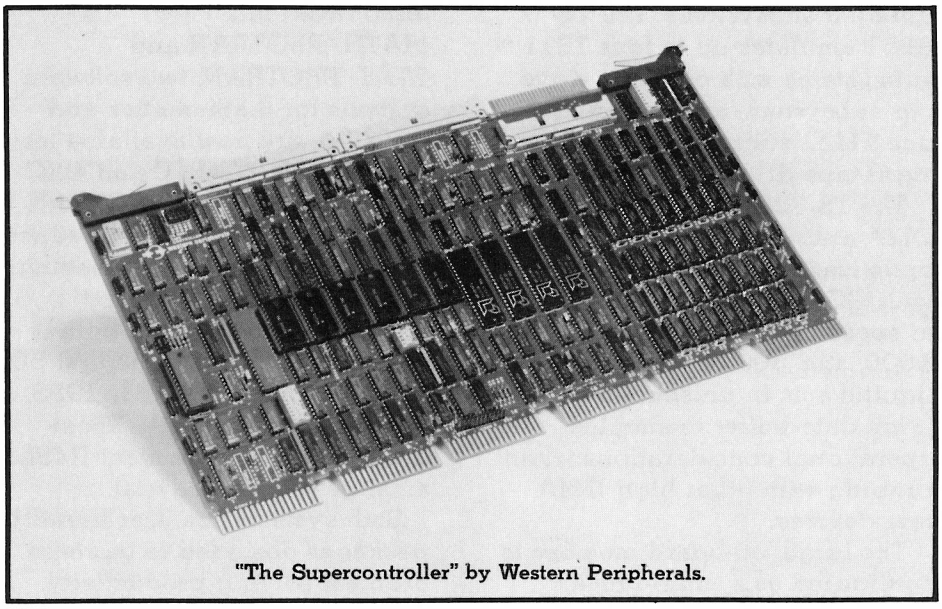
The Bus Isolation System solves the problem of modularizing a DEC bus. The inherent strength of a bus oriented computer system (the ability to attach most anything to it) is also its great weakness. If something goes wrong on the bus, it can take down the whole

system. The BUS protects against this problem by providing power and logic isolation from one subsystem to another, fault recognition within a subsystem, and either hardware or software controlled switching around the problem. Each BIS is an addressable device on the bus, and thus, each system can support up to 8 BIS subsystems.

To support the system's modularity needs, WRT has also developed the modular Power Controller System. Available in 30 amp and 50 amp versions, the PCS has four switched circuits for system use, and one unswitched circuit for test equipment use.

The FTS Systems are fully DEC supported with the exception of the WRT components. The system prices range from $100,000 and up and deliveries are expected to begin after the first of the year. The MDS (priced at less than $7000), and PCS (less than $2500) subsystems are also available individually from the company.

For further information contact John Sutherland, Western Resource Technology, Inc., 2970-Q East La Palma Avenue, Anaheim, CA 92806. Or telephone (714) 630-7852, or TWX 910-591-1241.



"The Supercontroller" by Western Peripherals.

## Western Peripherals Has 'The Supercontroller'

Tustin, CA — Western Peripherals, a division of WESPERCORP, announced the introduction of "The Supercontroller" GCR/Streaming Tape Controller for DEC PDP-11 and VAX-11 Unibus computers.

Called the TS-6251, the new Supercontroller "is the first streaming mode controller that allows compatible operation under standard unmodified software," explained Jack Olson, vice president of marketing. This is accomplished with the use of a 64 Kbyte on-board memory which de-couples the drive mechanics from the computer software.

The versatility of the TS-6251 enables it to be used as DEC TS11 or TM11 emulator, 6250 bpi (GCR) controller, software compatible streamer controller, 6250 bpi streamer controller, or a single board embedded stop/start controller.

An important feature of the TS-6251 hardware is its dual emulation capability, allowing it to emulate DEC TS11 or DEC TM11 subsystems. This feature is employed by implementing different sets of firmware for

different subsystems. The TS-6251 emulates up to four TS11 subsystems with one tape drive per subsystem, and it emulates one TM11 subsystem with up to eight tape drives.

The TS-6251 has a unique DMA auto-throttle feature that optimizes the controller's use with GCR and tri-density drives at speeds up to 125 ips, at 6250, 1600, and 800 bpi. The auto-throttle acts in unison with the large data buffer to simplify operational considerations when running with other high DMA rate devices.

The large, on-board memory is configured as a multi-block staging buffer and enables the user to run ½" streaming 1600 and 3200 bpi drives with unmodified software in a streaming mode. The buffer accumulates multiple blocks of data to be written or read, and determines when to stop and start the drive, maximizing throughput. This new concept is highly cost effective as "memory chips have come down in price during the past year—allowing for greater storage on a controller board," said Olson.

"The Supercontroller will even do error retries without software intervention," he added. This eliminates most of the time-consuming repositioning cycles that make streaming drives unable to run unmodified software on other controller systems.

Price for the TS-6251 Supercontroller is $2250 in single quantities, including cables, documentation and diagnostics. Delivery is 30 days.

For further information, contact Jack Olson, vice president of marketing, Western Peripherals, 14321 Myford Road, Tustin, CA 92680. Telephone: (714) 730-6250.

## New Software Systems For VAX-11, DEC 10/11 Available From IMSL

Houston, TX — IMSL, Inc. has announced that MATH/PROTRAN and STAT/PROTRAN, two software systems for mathematics and statistics, are now available for the VAX-11, DEC 10 and DEC 20 computers. Both PROTRAN products are designed to reduce programming effort at a savings to the user.

A free 60-day trial of both systems is offered to the first 500 users or until March 31, 1983, whichever occurs first. Those interested should contact IMSL to arrange for the free trial.

Both systems are user-friendly packages designed to increase problem-solving productivity. Advantages to the user are that no formal programming knowledge is needed for its application; interface between products is compatible; error checking is provided and *Fortran* can be easily intermixed with the more powerful PROTRAN statements.

MATH/PROTRAN solves problems in elementary operations and random number generation, interpolation and data smoothing, integration and differentiation, differential equations, linear and nonlinear algebraic equations, eigenvalues and eigenvectors, optimization, transforms and sorting.

STAT/PROTRAN assists in problem solving areas such as basic statistics, frequency tables and crosstabulations, correlation, regression analysis, analysis of variance, and random number generation.

Initial annual subscription rates for the DEC VAX-11 Series are MATH/PROTRAN - $3,000; STAT/PROTRAN - $2,500. The subscription rate for both MATH/PROTRAN and STAT/PROTRAN is $4,500 the first year. IMSL is offering an introductory university price of $500 for initial subscriptions for each product.

For additional information, including details on the free 60-day trial, contact IMSL, Inc., Sixth Floor, NBC Building, 7500 Bellaire Blvd., Houston, Texas 77036-5085 USA. Telephone (713) 772-1927. Outside Texas, call toll free (800) 231-9842, or telex 79-1923 IMSL INC HOU.

## Evans Griffiths & Hart Offers RSTS/E Package

Lexington, MA — Evans Griffiths & Hart, Inc. (EGH) has announced the release of a new product, VMSPIP, a RSTS/E package that reads disks written under VAX/VMS and copies VMS file to RSTS/E media.

Describing VMSPIP as extremely fast, an EGH representative said the package was designed for those installations where tape-based file transfers from VAX/VMS to RSTS/E were too slow and unwieldy to be practical. He added that the package is particularly convenient where a VAX and a PDP-11 share a switchable dual-ported disk.

In addition to preserving the normal RMS attributes of a VMS files, VMSPIP can also recognize and properly copy files created by ROSS/V, EGH's RSTS/E Operating System Simulator under VAX/VMS. This allows a PDP-11 to provide operational backup for a VAX that is running critical applications under ROSS/V.

A single-CPU license for VMSPIP is $1250. OEM and quantity discounts are available. Contact Evans Griffiths & Hart, Inc., 55 Waltham Street, Lexington, MA 02173. Tel: (617) 861-0670.

## SOFPROTEX Releases Tape Library System

Belmont, CA — SOFPROTEX, a division of Government Copyright Services, has just released DATASAFE, a tape library system which performs a wide variety of functions to aid

"*The Bridge*™ is software that creates a virtual microcomputer at every terminal connected to my mini. I have all the functions of a micro, but without micro limitations.

"The *z-Board*™ has four z-80a® microprocessors per board to execute programs at high speed. Faster than many dedicated micros. And it has 256K bytes of RAM, plus a bit slice state machine. That's the guts of four micros for less than you might pay for one.

"With *The Bridge*, I can run CP/M® based programs. I like that. And micro programs like Supercalc® are easy to use, and inexpensive. I like that, too.

"But the best thing about *The Bridge* is systems integration. Now everyone in the office uses the same system — no more problems with disk formats, incompatible languages or programs. *The Bridge* provides each user with a virtual microcomputer with the advantages of a mini's high-speed printers, hard disks, and communications.

"*The Bridge* with a *z-Board* gives me the performance of four microcomputers — at a fraction of the cost."

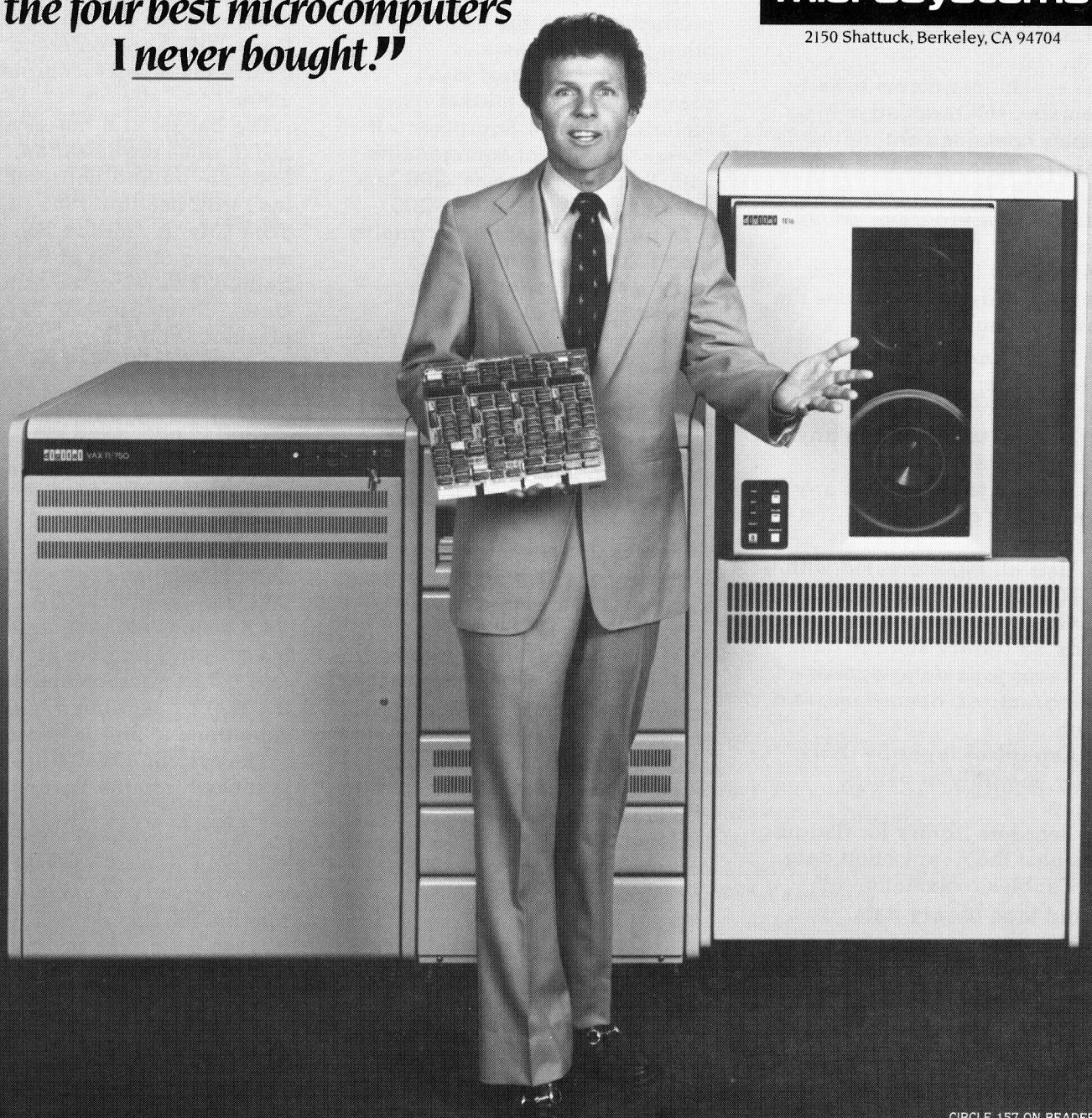
The Bridge and z-Board are trademarks of Virtual Microsystems, Inc.
z-80a is a registered trademark of Zilog, Inc.
CP/M is a registered trademark of Digital Research.
Supercalc is a registered trademark of Sorcim, Inc.

For information, call Jim Swanson (415) 841-9594.

**virtual microsystems**

2150 Shattuck, Berkeley, CA 94704

# "The Bridge and a z-Board — the four best microcomputers I never bought!"



CIRCLE 157 ON READER CARD

the user in the restoration of lost or corrupt data. The system runs under RSTS/E, version 7.0, 7.1, or 7.2.

DATASAFE is capable of maintaining from 1 to 7 separate computer system tape libraries and is currently used by Apple Computer to keep track of PDP 11/70, VAX and IBM System 38 tapes at 3 different sites.

DATASAFE runs under the Basic Plus run-time system and keeps track of tape information utilizing a user-friendly interface. This menu-driven software product comes with full documentation and is easy to install.

The system maintains orderly, systematic, self-tracking of tape statistics contained within the library for reliable re-call and verification. This process is internal to the program, going unnoticed by the operator. Among more than 20 diverse features, the system provides the following functions:

1. Stores and retrieves information on any single tape or volume set.
2. Lists all tapes having any given label name.
3. Lists all tapes having any given date of creation or expiration.
4. Lists all tapes created within any 10-day time-span.
5. Generates several formatted reports on library storage data.
6. Keeps tape data segmented by programmer, operations, and system.
7. Organizes tapes by daily, weekly, monthly or yearly backup.
8. Reorders library for flagging tapes past their expiration date.
9. Enables reclamation of expired tape library data.

SOFPROTEX supplies DATASAFE complete with software to generate tape labels for rack storage. For more information write to: SOFPROTEX, P.O. Box 271, Belmont CA 94002 U.S.A.

## Southern Systems Offers 140 CPS Matrix Printer

Fort Lauderdale, FL — Southern Systems Inc., printer system specialist, has announced a new 140 CPS (character per second) matrix printer with a plotting format option.

The M-100 matrix printer offers output at 56 lpm when printing 132 characters per line, at 96 lpm when printing 72 characters per line and 145 lpm at 40 characters per line.

The new Southern Systems product, as with the company's other printer systems, is compatible with most mini computers on the market, including Digital Equipment. It shares 70 percent commonality with Southern Systems' 200 lpm dot matrix printer, the M-200.

Designed for near letter quality the M-200 offers options for graphics, bar codes, programmable character generator and block letters. With the block-letter option, characters may vary in height from .08 inches to 0.8 inches, well-suited for bold headlines or easily readable labels. The bar code option requires a minimum of data from the host CPU. Graphics options allow use of bar graphs, X-Y plots and special symbols and designs. The user-programmable 128-character generator allows users to print with two languages or font styles; special characters are easily programmed.

The forms handled by the M-100 are continuous fanfold, edge perforated from three to 16 inches wide. Up to six copies (original plus five) can be produced.

Southern Systems is a full-service printer system specialist offering end users a wide range of impact and non-impact devices that range upward to 5,280 lines per minute.

For more information write Southern Systems, 2841 Cypress Creek Road, Fort Lauderdale, FL 33309, or call (305) 979-1000; (800) 327-5602.

## Software Ireland Enters U.S. Market; Gordon Bell To Head New Organization

New York, NY — Software Ireland Ltd., a three-year-old Belfast company that is part of the ICS Computing Group, plans to enter the U.S. market this year with a new commercial programming language that will extend the use of software developed for Digital Equipment Corp. (DEC) computers to thousands of microcomputer users.

The Belfast firm has established a U.S. sales organization, Software Ireland Representatives, Inc., with headquarters in New York City. R. Gordon Bell, managing director of Software Ireland, has relocated to the United States and will serve as president of the new organization. He will personally supervise the building of a sales network throughout the country.

Bell said that Software Ireland's SIBOL language compiler was developed over the past year specifically to provide a link between the popular DIBOL language developed by DEC for use on its PDP-11 and VAX ranges of computers and the growing number of 16-bit and 32-bit microcomputers using the UNIX operating system developed at Bell Laboratories.

The system already has been launched into the European market, Bell said.

"The success we've had with SIBOL in Europe convinced us we were ready to make our move into the highly competitive U.S. market, where the potential growth of business applications software on UNIX-based microcomputers is phenomenal," he stated.

"We have identified almost 40 hardware manufacturers who

offer machines using UNIX, which means that a great many end users already exist who can benefit from the superb DIBOL software programs now being marketed exclusively to DEC users."

Bell said he is moving rapidly to establish a sales network to market the SIBOL system to computer manufacturers, OEMs, distributors, software suppliers, and possibly even direct end users.

SIBOL had its formal introduction to the U.S. computer industry at the November, 1982, COMDEX trade show. The system consists of a compiler, a run-time interpreter, a symbolic debugger, and a library of eternal utility subroutines.

Bell said the company will offer non-exclusive licenses for the compilers and run-time systems to manufacturers and distributors. He estimated that prices will be on the order of $2,000 for the compiler and $500 for run-time licenses.

Software Ireland is a wholly-owned subsidiary of National Westminster Bank, one of the world's 10 largest banks, and is part of its ICS Computer Group, the largest computer services company in Ireland. Formed in 1979, Software Ireland now has more than 30 employees at offices in Belfast and Dublin.

For further information contact: R. Gordon Bell, Software Ireland Representatives, Inc., 100 Wall Street, New York, NY 10005; (212) 509- 0363.

## Emulex Adds Capability To CS 11/21 Multiplexers

Santa Ana, CA — Emulex Corporation has added two products to its line of communications equipment, William Dollar product manager, Communication, recently announced.

The two products, called the CS11/F and CS21/F, add DMF-32 capability to Emulex's CS11 Series and CS21 Series of communications multiplexers.

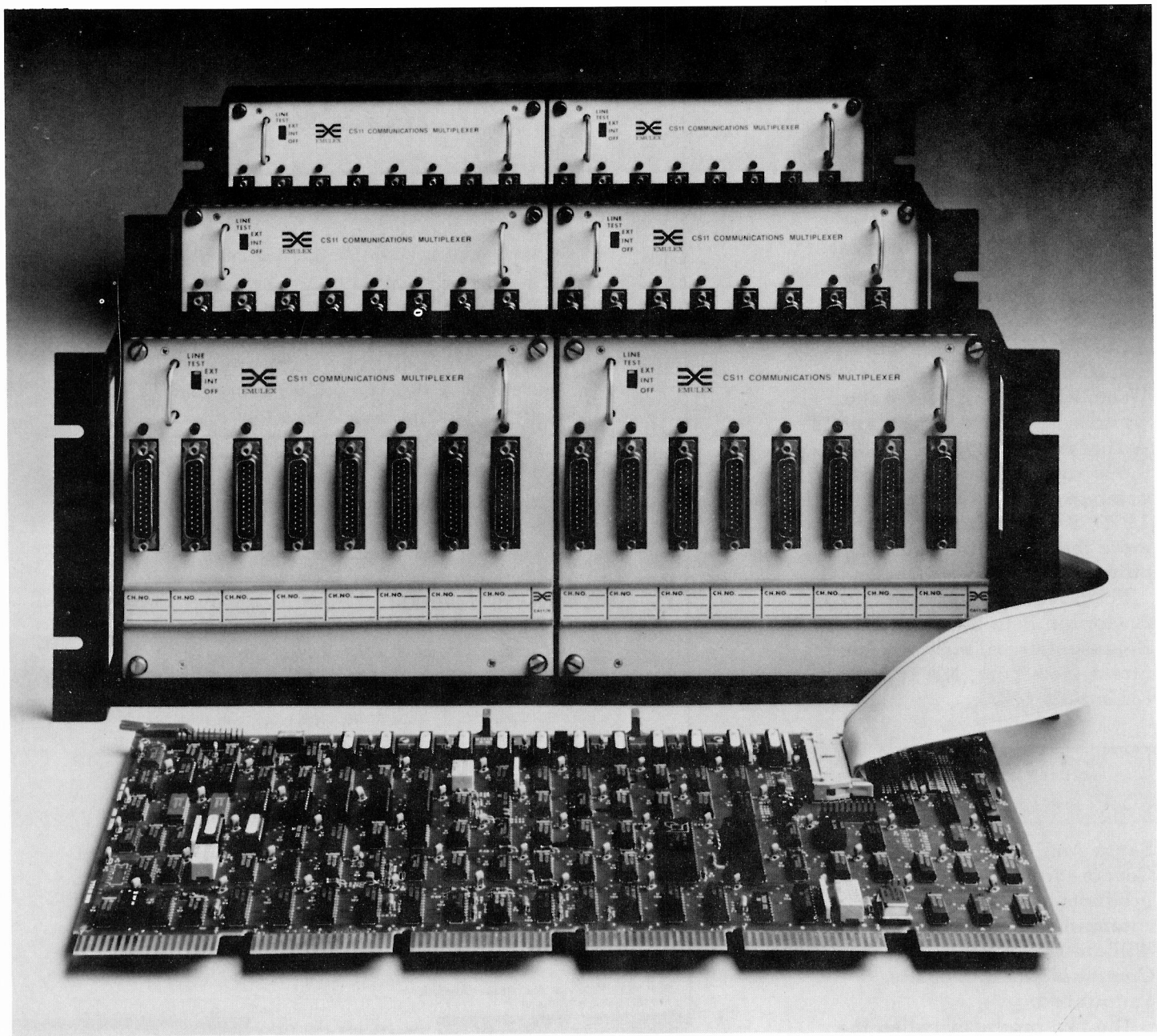
Both models are fully software transparent and operate in the VAX-11 environment with VMS 3.0 and above. Also, they are fully software transparent to the stand-alone and on-line versions of DMF-32 diagnostics, meaning that these products are at the disposal of all VAX users, including the 11/730, 750, and 780 CPU's.

"The major advantage of the DMF-32 emulation," said Dollar, "is the significant performance improvement gained through the new, highly optimized terminal handling software in VMS. These Emulex products permit the user to realize this same performance—plus gain the many added advantages offered

by the CS11 and CS21 hardware, all at considerably lower cost.

"One major advantage of these Emulex products over the DMF-32," continued Dollar, "is in the area of modem control: The DMF-32 offers eight asynchronous lines, one synchronous line, and one parallel I/O port. But, only two of the eight asynchronous lines have modem control needed for remote telecommunications. The other six lines are for local use only. The CS11/F and CS21/F respond to the real need of users, most of whom simply need to concentrate many asynchronous lines into their VAX CPU. Therefore, these products are ideal to implement new systems and also as add-ons to a system already equipped with a DMF-32. Both models incorporate modem control on all lines. Thus, any or all lines (in any combination) may be used for any mix of local or remote lines.

"Another big advantage of the Emulex products," added Dollar, "is their efficient, compact packaging. The CS11/F, for example, handles as many as 48 asynchronous lines on a single board — the equivalent of six DMF-32 boards. All lines have full modem control and may be used in either full- or half-duplex modes. A fully configured CS11/F system consists of a single host-mounted hex-sized controller board and three external distribution panels of 16 ports each, making it ideal for applications involving large numbers of local or remote terminals and/or where power




The F model brings DMF-32 capability to Emulex's popular CS11 Series of communications multiplexers.

and backplane space are at a premium."

The CS21/F offers a maximum capacity of 16 aschyronous lines on a single hex-sized board — the equivalent of two DMF-32 boards. All lines have modem control and are used in the full-duplex mode only. A fully configured CS21/F system consists of a single host-mounted controller board and a passive 16-line distribution panel. The CS21/F is ideal for applications involving a smaller number of local or remote terminals and where backplane space and power requirements are not critical.

"Pricing is exactly the same in both the CS11 and CS21 Series as for our current DH11 emulations," said Dollar. There is no premium paid for the higher performance. For example, a 16-line CS11/F lists for $4500, and a 16-line CS21/F lists for $3500. Attractive discounts are available to OEM and volume users on a mix-and-match basis with all other Emulex products. Plus, existing CS11 and CS21 owners can convert their configurations to the DMF-32 version by PROM changes for $350."

The CS11/F has been available in quantity since Jan. 1, 1983; the CS21/F since Jan. 15, 1983.

Like all Emulex products, the CS11/F and CS21/F are constructed of pre-tested and pre-aged parts and are tested at least twice at a complete subsystem level. Each unit is environmentally cycled from 40 to 125 degrees F for 96 hours prior to shipment to minimize infant mortality failures. The products are backed by Emulex's standard one year warranty and by the Company's extensive service network.

For further information, please write or telephone Phillip Begich, director of national sales, Emulex Corporation, 2001 East Deere Avenue, Santa Ana, CA 92705; (800) 854-7112, or in California (714) 557-7580.

## GEJAC Announces ARSAP for RSTS

Riverdale, MD — The ARSAP Resource Management and Chargeback System, has been updated to run on PDP-11 computers using RSTS versions 7.0, 7.1 and DEC's recently released 7.2 operating system.

ARSAP is a comprehensive computer resource management system which provides users, system managers, corporate and financial staff with reliable reports for making informed, accurate decisions about the data center operation.

ARSAP is now available on all the most widely used DEC operating systems: RSTS, RSX-11M-PLUS and VAX/VMS. Managers of multiple operating system shops can get compatible reporting of resource utilization, making it easy to compare productivity on all in-house DEC systems. This compatibility has not been available before now.

The most significant features of ARSAP are:

1. Project Accounting — Users can be prompted for a project-id and password upon logon and the terminal session can be allocated to his project as well as his logon account, so it is easy to monitor projects and users.

2. Terminal Accounting — System utilization reports can be generated for each terminal, showing the activity on each terminal, when the terminal is used and the amount and type of resources consumed. The reports are presented in graphic as well as numeric format.

3. System Utilization — Utilization reports can be generated by shift and for any user-selected date range. Reports include information on CPU usage, disk space, number of logons, connect time, device time, printed lines, and other major resources.

4. Invoicing — Invoices can be generated by user or by project. Rates can be by time of day and can be set by individual user, project, or system default basis.

ARSAP is used to control day to day operations, to monitor system utilization and growth, to plan the resources needed for current and future workloads, to justify needed hardware and software upgrades, and to prioritize and schedule computer processing demands.

ARSAP also can produce itemized invoices automatically for allocating costs of computer resources to internal departments, projects, users, contracts and grants. It is used for departmental budgeting, to control project costs, or to bill timesharing customers. ARSAP's equitable billing system enables bidding competitively on contracts and satisfies government and commercial job costing requirements.

Multiple copy discounts are available. ARSAP is delivered within two weeks on mag tape, comes with a 30-day acceptance period and a full one year warranty, including maintenance and enhancements services, and complete documentation.

Contact: GEJAC, Inc., P.O. Box 188, Riverdale, MD 20737, 301-864-3700.

# PRODUCT UPDATES

## Emulex Reduces Prices On Many Unibus Products

Santa Ana, CA — Emulex Corporation has reduced prices on selected members of its family of Unibus disk and tape controllers, Roger S. Evans, director of product management, recently announced.

The price reductions cover Emulex's SC21/V disk controller and TC11/N and TC11/P tape controllers.

The new prices were effective November 1, 1982.

List price reductions are:

—The SC21/V, designed for Unibus use with DEC's VAX-11 series of computers has been reduced from $6000 to $5000, a decrease in price of 16 percent.

—The TC11/N, which cost $3000, now sells for $2200, a decrease in price of 27 percent.

—The TC11/P, formerly listing at $3600, now lists at $2800, a decrease of 22 percent.

The TC11/N is a single density NRZI tape controller. The TC11/P is a dual density tape controller that supports both NRZI and PE modes.

"These disk and tape controller price reductions reflect Emulex's improved manufacturing efficiency, and we have decided to pass these savings directly to our customers," Evans said.

He also pointed out that these new low list prices are further reduced for OEM and volume customers who take advantage of Emulex's product mix-and-match discounts. Under this program, all purchases from Emulex in any year — regardless of whether for disk, tape, or communications products — count toward gross discount credits.

For further information call or write Phillip Begich, director of national sales, 2001 East Deere Avenue, Santa Ana, CA 92705. Telephones: (800) 854-7112, or in California (714) 557-7580.

## Catch-23 Now Available On RSX-11M Version 4

Sudbury, MA — EEC Systems announce that their Catch-23 software is now available on RSX-11M version 4. Catch-23 is a software package which allows DEC PDP-11/23 users to upgrade from 18 bit to 22 bit addressing capabilities without having to replace existing 18 bit peripheral devices.

A company spokesman said that this represents a cost savings of several thousands of dollars over buying new hardware. He added that sales of Catch-23 have been brisk since the product was first announced last summer and has been installed at numerous Fortune 100 companies. Catch-23 is priced at $1995.00 for a single CPU license.

For more details contact: Eric Dickman, EEC Systems, Inc., 327/E Boston Post Road, Sudbury, MA 01776, (617) 443-5106.

## Solutions DECk Offers User Productivity Tools

Fredericton NB, Canada — A family of programmer productivity tools is now available for RSTS users from SOLUTIONS DECk.

The SOLUTIONS DECk is a family of products to aid in the quick and accurate production of the

# RSTS/E INTERNALS MANUAL

The RSTS community has been clamoring for years for a book that details the inner workings of RSTS/E. Well, clamor no more. Michael Mayfield of Northwest Digital Software, and M Systems, the publisher of The RSTS Professional and The DEC Professional Magazines, have teamed up to produce the RSTS/E Monitor Internals Manual.

This manual describes the internal workings and data structures of the RSTS/E monitor. It also notes differences in the internal structures between version 7.1 and earlier versions of the monitor. Future updates will include changes for new versions of the monitor.

Information is available for all levels of users:

- Gain a basic understanding of the workings of the monitor for optimizing system performance.
- Information on disk structures allows recovery of data from corrupted disk packs.
- Special uses of runtime systems and resident libraries allow complex applications to be developed without degrading system performance.
- Write your own custom device drivers for that "foreign" device you need to add but thought you couldn't.

CONTENTS:

Chapter 1 describes the structures used by the monitor that are resident on disk. These include the directory structure, disk allocation tables, Save Image Library (SIL) formats, bootstrap formats and bad block mapping.

Chapter 2 describes the tables used within the monitor to control system resources and provide program services. These tables provide job, memory, file and device control, as well as program services such as interjob communication.



Chapter 3 contains information on writing and installing a custom device driver. It describes the entry points and information the driver must provide to the monitor as well as the subroutines and macros the monitor provides for the driver.

Chapter 4 contains information that enhances information already provided by Digital on writing custom resident libraries and runtime systems. It concentrates mainly on non-standard uses of resident libraries and runtime systems to increase system performance and functionality.

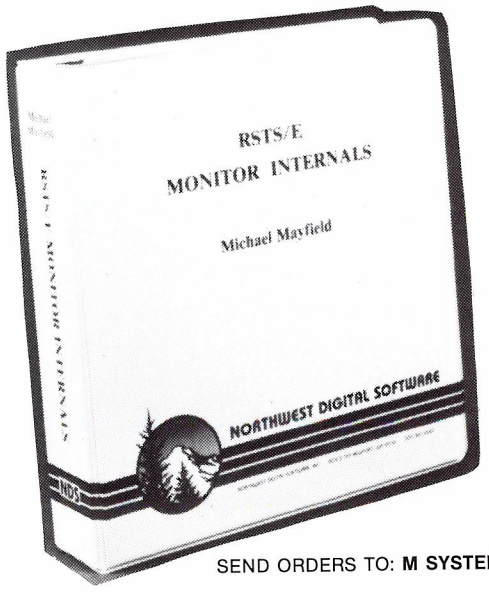
Appendix A provides six quick reference foldout charts:

- The directory structure.
- The monitor tables.
- Fixed memory locations and common data structures.
- Monitor subroutines.
- Device driver entry points.
- Device driver macros.

Appendix B provides examples of the peek sequences required to access most of the monitor tables. It also contains an example program that uses many of the monitor tables to display a job and open files status.

Appendix C provides an example device driver.

Appendix D provides an example runtime system that doubles as a menu system for restricting specified users to a menu of options.




**$95.00**

SEND ORDERS TO: **M SYSTEMS, INC., BOX 361, FORT WASHINGTON, PA 19034-0361**

most common types of commercial data processing programs.

SOLUTIONS DECk includes preprocessors to aid in producing report, inquiry, master and transaction programs. These individual, low cost tools should provide at least a 100% productivity improvement in even casual use.

Taking advantage of the similarities between programs of the same type, SOLUTIONS DECk will produce greater than 90% of the code required. The programmer is provided with the opportunity to include his own code for the necessary unique solution to the particular task.

SOLUTIONS DECk supports the use of Digital's Basic+2 and RMS-11 if desired. (Basic+2, RMS-11, DEC and RSTS are trademarks of Digital Equipment Corp.)

Introductory prices are all less than $800.

For more information write: P.O. Box 684, Postal Station A, Fredericton NB E3B 5B4, Canada; or call 506-455-1008.

## Digital To Distribute Virtual's 'The Bridge'

Berkeley, CA — Digital Equipment Corporation has become a distributor for Virtual Microsystems, a Berkeley based software developer. Virtual Microsystems has created a software product called **The Bridge**, which provides a virtual CP/M environment on DEC minicomputers. **Bridge** users can directly run CP/M applications on their DEC minicomputers; programs which are often very user-friendly and which are usually quite inexpensive.

DEC will also market Virtual's **PhoneLink** communications package, under the name "Bridge Communications." **PhoneLine** allows minicomputer users to transfer binary and text files between CP/M microcomputers and their minicomputers.

A user of **The Bridge** runs the program like any other task. Once invoked, **The Bridge** causes the terminal to act as though it were connected directly to a CP/M microcomputer. The only difference is that the system substitutes "virtual floppies" for the diskettes used by microcomputers, a virtual floppy being a file on the user's hard disk.

The virtual nature of **The Bridge** system is quite useful.

Southern System's new 140 cps (characters per second) matrix printer with a Plotting Format Option.

Instead of dedicating additional microcomputer hardware to new users, any existing terminal tied to the DEC minicomputer can function as a microcomputer. Moreover, CP/M files can be shared like any other minicomputer file, and are backed-up automatically with the rest of the minicomputer system.

In reality, **The Bridge** is an alternate approach to microcomputer networking. Instead of building a microcomputer network, with all the attendent problems of communications and incompatibility, **Bridge** users merely add CP/M capability to the existing DEC network. The CP/M operating environment and the thousands of CP/M application programs become one of the many functions available within the DEC environment.

**The Bridge** system can be supplemented by adding a hardware accelerator available directly from Virtual Microsystems. The accelerator, known as the **z-Board**, features four z80a microprocessors, 256Kbytes of RAM, and a bit slice state machine. Once installed, the **z-Board** works with **The Bridge** to execute CP/M programs at the full speed of a dedicated microcomputer. The integrated **Bridge/z-Board** system allows DEC users to get full microcomputer functionality within their existing networks at a fraction of the cost of adding microcomputers.

DEC is currently marketing the RSX and VMX versions of **The Bridge** for the full line of PDP/11 and VAX minicomputers. **Bridge Communications** will be available shortly.

## First DR11-W Module For LSI-11's From MDB

Orange, CA — After recently introducing a Direct Memory Access Module (DMA) for PDP-11 processors, MDB Systems, Inc. has now developed the only Q-Bus DR11-W for LSI-11 based systems.

Designated MLSI-DR11-W, it is compatible with DEC's operating software for the Unibus DR11-W with several added features. The MDB quad size module has a switch selectable 22-bit addressing mode, and Bus Address Extension (BAE) register per DEC format, that allows direct memory transfer throughout the 4 mega byte range, and MDB's exclusive DMA throttle feature so system designers can maximize their CPU capability.

In addition, it offers the exclusive design features of four level or single level interrupt arbitration (also switch selectable and is compatible with 16, 18, and 22-bit processors.

Like MDB's Unibus module, the Q-Bus MLSI-DR11-W is a high speed digital input/output device designed for use with high speed graphics, digital data acquisition, any application where parallel information needs to be processed quickly, or as an interprocessor link between a Unibus and Q-Bus system.

Other exclusive features of the Q-Bus DR11-W include self-test from on-board diagnostics which are micro sequencer driven, and ease of set-up and cabling. Edge mounted LED's indicate error conditions and proper performance.

The MLSI-DR11-W is available in 30 days ARO and is priced at $995 in single quantities.

For additional information contact the company at 1995 N. Batavia Street, Orange CA 92665; 714/998-6900.

## Digital Info. Systems Now Has DBL/VMS in Native Mode; Announces Combo Package with S&H

Sacramento, CA — Data Business Language (DBL) is now available in a VMS native mode version, according to Digital Information Systems Corporation.

DBL has been marketed since 1978; and with the addition of DBL/VMS, this DIBOL-11 source code compatible language and compiler is now available for RT-11, TSX/TSX-Plus (time sharing extensions to RT-11), RSTS, RSX-11M/M-Plus, VAX/VMS compatibility mode, and VAX/VMS native mode.

DBL/VMS features include:

1. The DBL/VMS compiler is written in VAX/VMS native mode and is a true compiler.
2. Output of the DBL/VMS compiler is in-line native code.
3. Multi-user programs can access shared XCALL libraries.
4. Entire applications can be "bound" into a single executable module (i.e., an Accounts Payable application).
5. Little or no modification is required to run existing CTS-300 DIBOL code under DBL/VMS native mode.
6. DBL/VMS uses the RMS file structure. Those files are then accessible to Datatrieve, FMS, and all other VMS supported languages.
7. DBL programs can access and be accessed by other languages.

List price is $5,300.00 and quantity discounts are available to OEM's.

In a related development, Digital Information Systems Corporation and S&H Computer Systems Inc. of Nashville, TN, are jointly offering a combination package that includes DBL, TSX-Plus, and RTSORT.

DBL is a structured superset of DEC's DIBOL and is source code compatible with DIBOL. DBL is currently available for RT-11, TSX-Plus, RSTS, RSX-11M, and VAX/VMS.

Price per package ranges from $1420.00 for a quantity of 5 down to $1154.00 for 100 or more.

For more information, please contact Digital Information

Systems Corporation, 3336 Bradshaw Road, Suite 340, Sacramento, CA 95827, or call (916) 363-7385.

## Auto-Dial Support For DMG/NET Package

Ontario, Canada — Digital Management Group Ltd. has announced that DMG/NET (their networking software for DEC PDP-11, RSTS/E systems) now interfaces with Racal-Vadic and Ven-Tel auto-dial units.

DMG president John Dightam points out that for some networking needs, auto-dial can be cost-effective. "DMG/NET is well established for X.25," he said, "but there are many data communications situations where auto-dial is a better choice. The often substantial cost savings of packet switching frequently don't apply to links in a 'local calling' area, and in some countries X.25 networks are very limited, expensive or unreliable."

In a typical DMG/NET configuration, an auto-dial modem is connected to one of a DEC computer's terminal ports. The software is configured for all required destinations, and the user can then access any destination by typing a short, meaningful abbreviation at the terminal. DMG/NET establishes the connection to the remote computer and if the main number is busy will automatically seek alternates.

Auto-dial and many other network "gateways" can all be included on the same DEC host. "This is the key to DMG/NET's ease of use," explained Dightam. "It's the 'Executive Work Station' concept, with a whole world of computing power brought to a single desk-top terminal. When a manager types NET NY, it's the New York computer he wants to talk to ... quickly and without concern for how the connection is made. An auto-dialer can also back-up the normal X.25 link, and keep the user happy even on days when the packets aren't travelling as they should!"

The Racal-Vadic and Ven-Tel auto-dial units support both 1200 and 300 baud, and DMG/NET automatically selects the speed appropriate to the destination. Both units have been tested by DMG, a fact which the company stresses as important.

John Dightam said "We designed DMG/NET to interface with almost any network unit or computer type that uses ASCII code, but there are some traps for the unwary. Nearly every 'black box' has its own quirks, and can sometimes require a special cable or chip. After all, auto-dialers were really designed to work with terminals, and allowing them to handle outbound calls from a DEC computer is a bit 'different'. We've had a lot of help from Ven-Tel and Racal-Vadic and their distributors, and now feel confident in recommending these units."

DEC, PDP-11 and RSTS/E are registered trademarks of Digital Equipment Corporation.

For suitable auto-dial model numbers, and further information on DMG/NET, contact Ken Allsopp, Digital Management Group Ltd., 4800 Yonge Street, Willowdale, Ontario, Canada M2N 6G5. Telephone (416) 225-7788.

## IB Graph Available To Digital Users

Placentia, CA — IB Graph, a menu-driven graphics software package which provides users of DEC PDP-11 and VAX systems with business graphics capability, is now available from Data Processing Design (DPD), Inc. The new package allows non-technical users to create bar, line and pie charts on a variety of graphics output devices.

"Not only can a relatively unsophisticated user create a variety of business graphics with IB Graph software, he or she can do it using existing data, or data entered on IB Graph's own complete data editor," said Eric Moothart, DPD's software marketing director.

IB Graph allows users to enter data for building graphs interactively on-line from a terminal, with the computer constantly monitoring and verifying operator command. Users can also process data in other formats, such as data processing files like RMS data management service, or list processing documents created through WORD-11 or DECWORD.

"IB Graph stores the definitions of charts and lets the user recall and modify these definitions to create exactly the chart needed," Moothart explained. "Charts that once took as long as two weeks to develop can now be produced in a few hours."

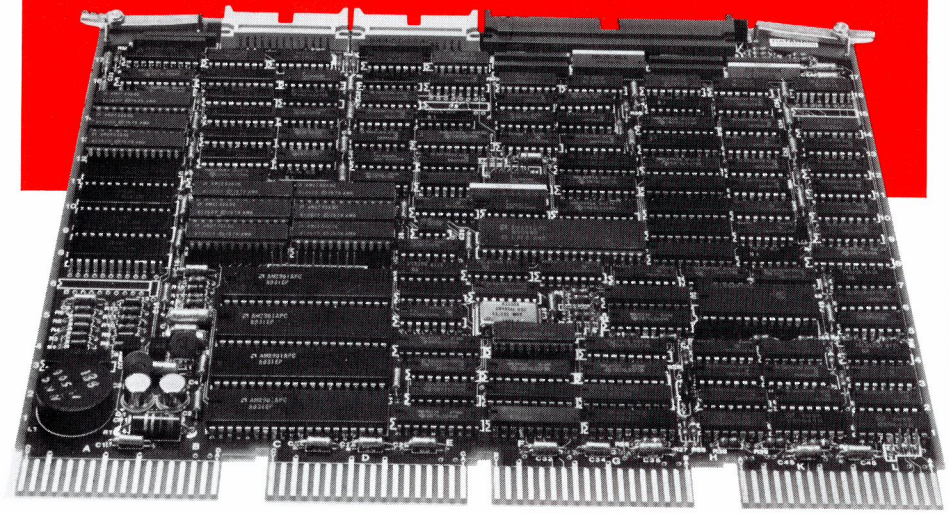
Using IB Graph users can enter and modify data and chart specifications on Digital VT100 (with advanced video option — AVO,) VT102 and VT125 terminals. The VT100/AVO command terminal will display only bar charts, while the VT125 will display bar, line and pie charts. When an appropriate color monitor is attached to the VT125, users can display their charts on the monitor in four colors.

IB Graph is implemented using Macro-11 and VAX BASIC for VAX/VMS and can be used on any valid VAX/VMS configuration — version 2.3 or later — with a working set size of 200 pages and at least 2000 free blocks of disk space.

As a multi-user interactive business graphics system, IB Graph operates in a timesharing environment under the RSTS/E, CTS-500 and VAX/VMS operating systems.

IB Graph is menu-oriented and includes a complete set of help screens to assist the user. The package consists of several

# DEC®-COMPATIBLE PERIPHERAL CONTROLLERS



**LSI-11® compatible controller for 80-300MB SMD and Winchester drives from CDC, Ampex, and Fujitsu**

Dataram Corporation offers the industry's widest range of DEC-compatible peripheral controllers — from comparatively simple NRZI tape controllers to complex 300 MB storage module drive (SMD) controllers.

An impressive array of state-of-the-art controllers, all built around high-speed bipolar microprocessors. All software compatible with the host LSI-11, PDP®-11, or VAX® minicomputer...and all available now.

And Dataram's controllers are designed to save you money, and, more importantly, space — our controllers typically occupy half the space required for the comparable controller from DEC. Doing it with a level of performance that makes any member of this family worth looking at.

The chart shows our current family of peripheral controllers, growing every day. If you don't see the controller you need, we're probably working on it right now. Call us and discuss your requirements.

| CONTROLLER | DESCRIPTION | COMPATIBILITY |
|---|---|---|
| C03 | Cartridge disk controller | RK05 |
| C33 | Cartridge disk controller | RK05 |
| T03 | NRZI mag tape controller | TM11/TU10 |
| T04/C | Mag tape streamer coupler | TM11/TU10 |
| T04/N | NRZI mag tape controller | TM11/TU10 |
| T04/D | Dual density mag tape controller | TM11/TU10 |
| T34/C | Mag tape streamer coupler | TM11/TU10 |
| T34/N | NRZI mag tape controller | TM11/TU10 |
| T34/D | Dual density mag tape controller | TM11/TU10 |
| T36 | Dual density mag tape controller | TM11/TU10 |
| T34/T | GCR mag tape controller | TM11/TU10 |
| S03/A, S04/A | 80 MB/300 MB SMD controller | RM02/RM05 |
| S03/A1, S04/A1 | 80 MB/160 MB SMD controller | RM02 |
| S03/B | 80 MB/300 MB SMD controller | RK07 |
| S03/C | 200 MB/300 MB SMD controller | RP06 |
| S03/D, S04/D | 96 MB CMD controller | RK06 |
| S33/A | 80 MB/300 MB SMD controller | RM02/RM05 |
| S33/A1 | 80 MB/160 MB SMD controller | RM02 |
| S33/B | 80 MB/300 MB SMD controller | RK07 |
| S33/C | 200 MB/300 MB SMD controller | RP06 |
| S33/D | 96 MB CMD controller | RK06 |

Products printed in red are LSI-11 Bus compatible.
Products printed in black are UNIBUS® compatible for PDP-11 and/or VAX minicomputers.

DEC, LSI-11, PDP, UNIBUS and VAX are registered trademarks of Digital Equipment Corporation.

**DATARAM CORPORATION**

Princeton Road
Cranbury, New Jersey 08512
Tel: 609-799-0071 TWX: 510-685-2542

CIRCLE 55 ON READER CARD

modules:

1. Main Menu module displays a list of available commands and also displays on request an index listing the names and reference numbers for data groups and chart definitions

2. Data Group Editor module allows the user to enter and edit data. This includes numeric data and label information for charts. With the editor, users can move, condense and manipulate data in various ways

3. Chart Specification Editor module provides complete control over the details of a chart, including type, colors, labels, etc. The user can select from more than 100 specific specifications to control the details of a chart

4. Plot module uses a data set and a chart specification to produce an actual chart on a device specified by the user

5. Utility module allows the user and manager to define devices that will be used to produce charts and assign names and characteristics to these devices

6. Data Transition module converts data processing files of various types (RMS, block I/O, ASCII, and others) and WORD-11 list processsing files into IB Graph data groups so they may be easily edited and plotted.

The price of IB Graph for users operating under VAX/VMS is $9500 for the PDP-11/780, $8500 for PDP-11/750 and $7500 for PDP-11/730. Secondary licenses cost $5500, $5500 and $4500 respectively.

Price of the IB Graph for users operating under RSTS/E is $7500 for the PDP-11/44 and $5500 for all others. Secondary licenses are $4500, $4000 and $3500 respectively.

Deliveries will be made 30 days ARO. IB Graph is distributed on magnetic tape compatible with nine-track 800 or 1600 bpi drives.

For more information, call DPD at (714) 993-4160. ♥

# CLASSIFIED

Send Classified Ads to: RSTS Classified, P.O. Box 361, Ft. Washington, PA 19034-0361.
$1.00 per word, first 12 words free with one year's subscription. [Be sure to include a phone number or address in your message.]







DEC 11/70 RSTS 192KB, RP04 DISK, TU10 TAPE, DH11 16 Channel, BA11F Expansion Box, Power Conditioner $60,000. (714) 827-1406.

ADM5 TERMINALS. Used 2 months. $490. Call Sara Cooperrider (202) 636-3000, x48.

STUDENT PROGRAMMER looking for part time position. Fairfield/New Haven County Area Connecticut. (203) 929-7149.

SAVE MEMORY with TYPE - 1K RTS, 1K user. Efficiently output disk files to terminals. 800/1600 BPI magtape, $100.00. ($150.00 after 31-Mar-82) Erskine Software, Box 86, Due West, SC 29639, (803) 379-8816.

Buy, Sell, Trade: DEC Systems, Parts, Peripherals. Call Paul, Digital Computer Exchange, Inc., 27892 Adobe Court, Hayward, CA 94542. (415) 886-8088.

MCBA CTS300, CTS500 Application software packages 50% off. Word-11 20% off. CTS300 license $1500. RIMS $2500. Serban Constantine, American Management & Information Services, (717) 496-7548.

## List of Advertisers

# For DEC* Computers, 100% Compatible.

# For DEC Owners and Users, 100% Essential.

**Thousands of New Products. 100% DEC-Compatible.**

All the latest hardware, software, services and supplies designed to run on your DEC computer. All the DEC-compatibles you've read about, heard about, but have never seen demonstrated. Plus, thousands more. Newer and better than anything on the market today. More DEC-compatibles than at any other show in the world!

*DEC and DECUS are registered trademarks of Digital Equipment Corporation.

**Over 250 Vendors. 100% DEC-Friendly.**

Meet the vendors who can help your DEC system reach a new standard of performance. Because you never have to ask, "Is it DEC-Compatible?" you get fast answers to the really important questions. It's the one Show for everyone who owns, manages, or uses a DEC computer. So bring the entire decision-making team: top management, financial management, DP management and senior staff.

Group Registration Discounts Available

**Bring Your Boss for FREE!**

Mail Coupon for Complete Details

**Especially for DECUS* Conference Registrants. 100% Free.**

It's easy to attend DEXPO East 83. And hard to miss. Especially for DECUS members attending the St. Louis conference. Free shuttle buses will take you from the Show to the conference in just five minutes. And conference attendees will be able to use their DECUS badges to enter the Show without paying a registration fee.

# DEXPO™ East 83

**The Third National DEC-Compatible Industry Exposition**

**Kiel Auditorium, St. Louis May 22-24, 1983**

MAIL TODAY FOR MONEY-SAVING REGISTRATION INFORMATION

☐ I want to save time and money on my DEXPO East 83 registration, air fare, hotel accommodations and car rental. Send complete information.

☐ Send _____ extra copies for my associates.

NAME _____

TITLE _____

COMPANY _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

☐ I am interested in *exhibiting* in the Show. Call me at

(____) _____.

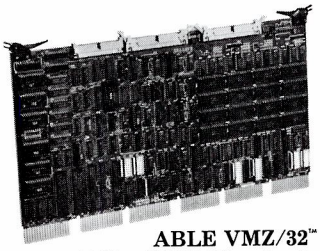
**Mail to:** Expoconsul International, Inc.
19 Yeger Road, Cranbury, NJ 08512
Tel: 609-799-1661

2 3

# If you're in the market for communications modules, make the ABLE connection now. And join the thousands who already have.

We are known as the innovators. Most of our products are industry "firsts" which become popular quickly, then settle into a stage of steady long-term acceptance. These four DEC-compatible, communications devices fit the pattern perfectly. They are ABLE originals. They achieved instant success worldwide. They provide top performance. And they are very reliable. Read on to find the one for you.

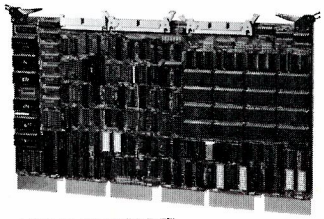
**INCREASED VAX THROUGHPUT.**



**ABLE VMZ/32™**
**16-line DMF/32 subset**

Here's an asynchronous microcontroller with programmable DMA, fully transparent to VAX/VMS as two 8-line DMF 32's and contained on a single board. Priced below the DZ11-E, it outperforms DZ or DH devices under VMS v.3, has interrupt-driven modem control on every line, and includes an output throttle which lets peripheral devices optimize their own data rate.

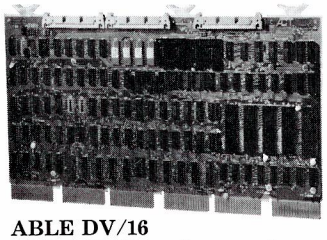
**#1 UNIBUS DMA.**



**ABLE DH/DM™**
**16-line combination DH11**
**& DM11 replacement**

Then there's our DH/DM, the original multiplexer which puts 16 lines with modem control on a single board. This popular device meets UNIX VAX system needs for DMA communications requirements, serves UNIBUS systems equally well, and beats them all for MTBF, throughput and price. Other features include on-board diagnostics, modem control on all lines, superior on-board silo depth and variable prom-set.

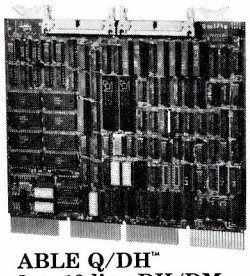
**SYNC/ASYNC FLEXIBILITY.**



**ABLE DV/16**
**16-line DV11 replacement**

A controller for the PDP-11 user, the DV/16 contributes microprocessor-derived flexibility, which permits mixing of sync and async lines in combinations of 4 or 8 lines with modem control and full system software compatibility. It takes less than half the space of a DV11 and uses word transfer instead of byte DMA to gain a 2 to 1 speed advantage or permit operation in half the bandwidth required for data transfers.

**Q-BUS DMA.**



**ABLE Q/DH™**
**8 or 16-line DH/DM**
**for Q-BUS**

The Q/DH is an asynchronous controller which makes DH-class performance possible on PDP-11/23 and LSI-11/23 Q-BUS systems. It connects the standard Q-BUS to as many as 16 async lines with DMA output capabilities and allows optimum Q-BUS utilization. Features include software compatibility with RSTS/E and RSX operating systems, large input silo, modem control on all lines.

Write for details on our complete line of DEC-compatible products. Be on the lookout for exciting new ABLE communications products soon to come.

## For Immediate, Toll-Free Information, Dial 800 332 ABLE.

ABLE COMPUTER

**CORPORATE OFFICES**
ABLE COMPUTER
1732 Reynolds Avenue
Irvine, CA 92714 • (714) 979-7030

**NATIONAL OFFICES**
Burlington, MA (617) 272-1330
Irvine, CA (714) 979-7030
Daly City, CA (415) 755-6040

**INTERNATIONAL OFFICES**
Canada (Toronto) (416) 270-8086
England (Newbury) (0635) 32125
W. Germany (Munich) 089/463080


# IB Graph for DEC users. If it wasn't so interactive, it wouldn't be used.

At Data Processing Design, we do one thing very well. Better than anyone else, in fact. We provide quality software for Digital Equipment Corporation computer systems. We call it Used Software, and it's the most thoroughly tested, well thought-out, and debugged family of software in the DEC-compatible industry.

Our latest product is IB Graph,™ the complete system for business graphics on DEC PDP-11™ and VAX™ systems. It's a multi-user, interactive business graphics system that lets you quickly and easily prepare a wide variety of charts—including pie charts, bar charts, and line charts.

IB Graph lets you enter data interactively or convert data from other formats, such as data processing files (including RMS files), or list processing documents (created through WORD-11™ or DECWORD™). Now you can store the definition of charts and recall and modify these definitions to create exactly the chart you need, when you need it.

And IB Graph is easy to use. Anyone can produce charts in a few minutes that used to take days to develop. It's menu-oriented and includes hundreds of help screens. Furnishing you with the kind of instant direction you'll need while learning IB Graph's operations. Which won't take long, either.

So if you'd like to know more about IB Graph, call or write to us at DPD. We'll be happy to provide you with all the information you need on our newest Used Software product.

And how using it to make your company more productive is as easy as, well, pie.

IB Graph

Data Processing Design, Inc.
AUTHORIZED digital® COMPUTER DISTRIBUTOR

CORPORATE OFFICE • 181 W. Orangethorpe, Suite F • Placentia, CA 92670 • 714-993-4160 Telex 182-278



CIRCLE 59 ON READER CARD